

## VI

### COMPUTACIÓN NEURONAL AVANZADA: FUNDAMENTOS BIOLÓGICOS Y ASPECTOS METODOLÓGICOS

José Mira; Ana E. Delgado  
*UNED*

#### 1. INTRODUCCIÓN

La computación neuronal nace hacia 1943 cuando W.S. McCulloch y Walter Pitts [McCulloch y Pitts, 1943; McCulloch, 1965], ante la evidencia de la calidad del tejido nervioso para procesar información y almacenar conocimiento en su estructura anatómica, proponen un modelo de neurona formal basado en la lógica al que actualmente llamaríamos circuito secuencial mínimo, formado por una función combinacional seguida de un retardo.

Actualmente [Anderson y Rosenfeld, 1989; Anderson y col., 1989; Fahlman y Hinton, 1978; Gallant, 1993; Hecht-Nielsen, 1988; Klimasauskas, 1989; Mira y col., 1993a; Rumelhart y col., 1986], se entiende por computación neuronal al cálculo realizado por arquitecturas modulares con un gran número de elementos de cálculo con un alto grado de conectividad y realizando localmente una función analógica no lineal (suma ponderada seguida de una función de decisión tipo umbral o sigmoide). La característica diferencial de este modelo de computación es que la programación externa se sustituye por el aprendizaje a través del entrenamiento usando algoritmos supervisados (refuerzo y retropropagación del gradiente, entre otros) y no supervisado (asociación espacio-temporal de estímulos) [Fahlman y Hinton, 1978; Gallant, 1993; Lee, 1991; Lerma, 1991; Prieto, 1991]. Esta visión de la computación neuronal presenta, al menos, los siguientes problemas [Mira y Delgado, 1991a; Mira y col., 1992, 1993c, 1994]:

1. Se aleja de forma prácticamente irreversible de la realidad biológica que le dio origen usando módulos de computación local claramente insuficientes para describir lo que en la actualidad conocemos sobre las neuronas reales.
2. No incluye autonomía en los procesadores, por lo que casi siempre es necesario un computador convencional para implementar los algoritmos de aprendizaje.
3. Hay una falta casi absoluta de metodología y un exceso de empirismo en los procesos de síntesis.
4. Se presenta como alternativa no integrable con la Computación Simbólica y la Inteligencia Artificial.

El propósito de este trabajo es contribuir parcialmente al planteamiento y solución de estos problemas. Para ello hablaremos primero de las analogías y diferencias entre la computación neuronal y la computación simbólica poniendo de manifiesto que toda computación es conexionista (neuronal) en el dominio propio de sus procesadores (puertas lógicas NAND o NOR, sumadores seguidos de sigmoideas o neuronas biológicas en el caso de los sistemas nerviosos). El simbolismo de «la otra» computación nace en el dominio del observador externo (programador) al tratar de reducir el planteamiento de un problema desde el nivel de conocimiento (lenguaje natural más metalenguaje del dominio) hasta representaciones directamente computables usando las primitivas de un lenguaje (por ejemplo C ó LISP). En un futuro, la computación neuronal también dispondrá de lenguajes intermedios de especificaciones funcionales. Si no lo consigue, es probable que desaparezca como rama de la computación.

El siguiente apartado de este trabajo está dedicado a introducir las bases anatomo-fisiológicas de la computación neuronal, buscando las exigencias computacionales mínimas de un modelo de neurona artificial que se aproxime a las neuronas biológicas algo más que los modelos usuales.

Está claro que cada uno busca inspiración donde puede, de forma que si definimos la computación neuronal como **modular, autónoma y autoprogramable**, no sería en principio imprescindible mirar hacia la Neurociencia para proponer buenos modelos de neuronas artificiales. Ahora bien, dado el origen del campo creemos que la biológica sigue siendo una fuente de inspiración inagotable y muy valiosa porque resume (ontogenia y filogenia) la historia evolutiva del ser vivo en su lucha adaptativa con el medio. Por otra parte, algunos de los modelos más complejos obtenidos a partir de la biología han mostrado ser también los más eficientes en la solución de problemas técnicos. En otras ocasiones no ha sido así. Volamos por conservación de la cantidad de movimiento y no moviendo las alas como los pájaros.

Finalmente, en el tercer apartado de este trabajo introduciremos algunos aspectos metodológicos en el análisis y diseño de redes de neuronas artificiales

(RNAs). El punto central del argumento en esta tercera parte del trabajo es la distinción entre «redes ciegas» y «redes basadas en conocimiento». El punto de partida de toda RNA debe ser el conocimiento que se tiene inicialmente sobre el problema a resolver. Veremos además que la mayoría de los problemas que tienen solución usando redes pueden formularse genéricamente como una tarea de clasificación, de forma que una primera aproximación a las redes basadas en conocimiento será su formulación como clasificadores genéricos.

El mensaje implícito en todo el trabajo es doble. Por una parte se sugiere no volver a reinventar la rueda (1943, 196x, ...) dado que ni siquiera es necesario hacerlo. La computación neuronal bien entendida (basada en conocimiento, autónoma, localmente autoprogramable, con teoría y metodología) es un campo de trabajo enormemente rico y prometedor. Esta es la segunda parte del mensaje: el optimismo para esta rama de la computación que se presenta no como alternativa sino como claro complemento a la Computación Simbólica y a la Inteligencia Artificial (IA).

## **2. PERSPECTIVA HISTÓRICA**

Hay tres etapas históricas en el campo de la Computación Neuronal, separadas por intervalos temporales del orden de los 20 años. La primera etapa comienza en 1943, cuando Warren S. McCulloch y Walter Pitts [McCulloch, 1965; McCulloch y Pitts, 1943] introducen el concepto de neurona formal como elemento de lógica de umbral, seguido de retardo y con propiedades de refractariedad. Con la introducción por Manuel Blum de la interacción de aferentes [Blum, 1961], una neurona formal puede calcular cualquier función lógica de sus entradas de forma que computacionalmente puede sintetizar un autómata universal de dos estados. Así, una red de neuronas formales de McCulloch-Pitts es equivalente a una máquina de Turing de cinta finita [Kleene, 1956; McCulloch, 1965].

En esta primera época, la computación neuronal y la simbólica caminan juntas [Kleene, 1956; Neumann, 1956; Shannon y McCarthy, 1956; Widrow, 1962; Yovits y Cameron, 1960]. Los trabajos de J. von Neumann, N. Wiener, S.C. Kleene, O.G. Selfridge, J.T. Culberson, J. McCarthy, Alan Newell, C. Shannon, N. Shapiro, E.F. Moore, Ross Ashby, Donalds McKay, A. Utley y Seymour Papert, entre otros, son representativos de los desarrollos formales que primero aparecieron bajo el paraguas de la Neurociencia y después dieron comienzo a la Inteligencia Artificial, en 1956. Hay tres recopilaciones [Anderson y Rosenfeld, 1989; McCulloch, 1965; Shannon y McCarthy, 1956] que resumen el estado del conocimiento en esta época: (1) «Embodiment of Mind» [McCulloch, 1965], (2) «Automata Studies» (editado por C.E. Shannon y J. McCarthy en 1956) y (3) una recopilación reciente editada por J.A. Anderson y E. Rosenfeld en 1989 («Neurocomputing: Foundations and Research»).

En la década de los 60 ocurren sucesos positivos y negativos para la computación neuronal. Por un lado se consolida la teoría de redes con la introducción del perceptron por Rosenblatt [1959], las ADALINES por Widrow [1962], el aprendizaje de Nilsson [1965] y los algoritmos de análisis y síntesis de Redes Neuronales a nivel lógico de W.S. McCulloch y R. Moreno-Díaz [Moreno-Díaz, 1971; Moreno-Díaz y McCulloch, 1968; Moreno-Díaz y col., 1972]. Aparecen también otras formulaciones analógicas más próximas a la Física debidas al grupo de Eduardo Caianiello [1966] y estudios en autoorganización a partir del trabajo pionero de Yovits y Cameron junto a von Foersters, Minsky, Newell, Shaw y Simon, Pask, McCulloch y A. Burks [McCulloch, 1960; Yovits y Cameron, 1960].

En esta época J. García Santesmases creó en España el primer grupo de investigación en redes neuronales en la Facultad de Ciencias de la Universidad Complutense. Las primeras publicaciones españolas sobre modelos electrónicos de neuronas capaces de aprender a clasificar estímulos mediante Reflejos Condicionados aparecen en 1963 y sus autores son J. Garcia Santesmases y R. Moreno Díaz. [1963].

En 1965 entran en el grupo F. Rubio y J. Mira trabajando en visión artificial y aprendizaje en redes neuronales, en colaboración con el profesor A. Gallego de la Facultad de Medicina. Resultado de los trabajos del grupo se construye en Madrid la primera retina artificial [García Santesmases, 1969, 1970a, 1970b], claro antecedente de los trabajos de Fukushima [1980] sobre el neocognitron, y la primera red neuronal probabilística [Moreno-Díaz, 1971; Moreno-Díaz y col., 1972], antecedente analógico [García Santesmases, 1976; Mira, 1971; Mira y Fonseca, 1970; Mira y col., 1972] de las pRAM de Clarkson y Ng [1993].

Hemos comentado algunos de los sucesos positivos para la computación neuronal en la década de los 60. Los sucesos negativos más relevantes fueron las críticas de Minsky y Papert en «Linearly Unrecognizable Patterns» [Minsky y Papert, 1969] y en «Perceptrons» [Minsky y Papert, 1969] sobre las limitaciones de las redes tipo perceptron en el campo de las funciones que no son linealmente separables (problema del XOR) y en el fuerte crecimiento de la computación simbólica en los campos de la I.A. y los sistemas basados en conocimiento. Existen sin embargo en esta década y en la primera mitad de los 80 trabajos en el área de Memoria Asociativa, Aprendizaje, Modelado Biofísico, Visión y Decisión Cooperativa que señalan la necesidad de modelos más complejos de computación neuronal [Carpenter, 1977; Delgado, 1978; Fonseca y col., 1970; Grossberg, 1972; Kilmer y col., 1968; Kohonen, 1972; Mira y Fonseca y col., 1970].

Durante esta época aparecen en nuestro país grupos procedentes del inicial de G. Santesmases en Zaragoza (R. Moreno y F. Rubio), Madrid (J.L. Aligué, M. Fernández, J.M. Girón), Granada (J. Mira, A. Delgado, E.L. Zapata), Las Palmas (R.

Moreno de nuevo) junto a otros grupos autónomos en el Instituto de Cibernética (C. Torras) y la ETSI de Telecomunicación en Madrid (G. Fernández, D. Maravall, ...).

Llegamos así al fuerte renacimiento de la computación neuronal en su tercera época, en torno a 1985, que se inicia con las propuestas de Rumelhart y col. [1986]. La sustitución de la función umbral por una función derivable que permite la retropropagación del gradiente, junto con otros factores procedentes de las limitaciones del campo de los sistemas expertos han hecho renacer con fuerza el conexionismo. Un buen resumen del estado actual, hasta 1989, se encuentra en Klimasauskas («The 1989 Neuro-Computing Bibliography»). Algunos trabajos representativos de este renacimiento aparecen en [Anderson y col., 1989; Carpenter y Grossberg, 1992; Fahlman y Hinton, 1978; Hopfield, 1982; Hopfield y Tank, 1986; Hwang y col., 1987; Lippmann y col., 1991].

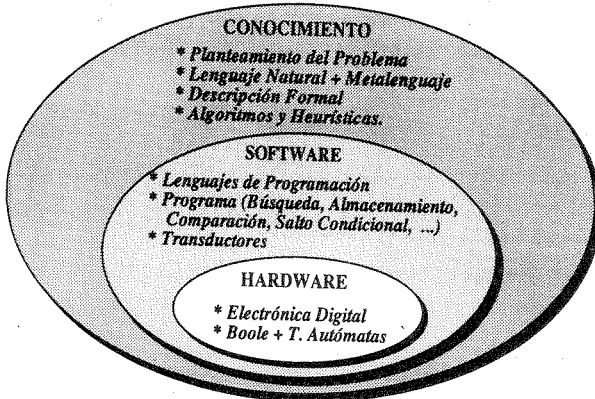
Dentro de esta última etapa de la computación neuronal se han desarrollado un conjunto importante de grupos de trabajo en nuestro país en prácticamente todas las Universidades. Una muestra significativa del trabajo de estos grupos queda reflejada en los capítulos de este libro.

### **3. ANALOGÍAS Y DIFERENCIAS ENTRE LA COMPUTACIÓN NEURONAL Y LA COMPUTACIÓN SIMBÓLICA**

Toda computación es conexionista (neuronal) en el dominio propio de sus procesadores (NANDs o sumadores seguidos de sigmoide). La distinción aparece entonces en el dominio del observador externo («programador») que ve la computación simbólica como una computación de «grano grueso» (estación de trabajo) que sólo necesita ser accesible a nivel de las primitivas de un lenguaje de programación de alto nivel. Sólo en circunstancias muy especiales se programa en ensamblador, cerca del lenguaje de los procesadores. En cambio, en la computación neuronal siempre operamos a nivel de procesadores analógicos.

De acuerdo con Alan Newell [Dietterich y Flann, 1989] el proceso fundamental en toda computación es reducir el planteamiento del problema desde el nivel de conocimiento hasta el nivel simbólico y posteriores niveles de implementación (software y hardware intermedios) hasta llegar al nivel de los procesadores.

En el paradigma computacional usual estos tres niveles (conocimiento, software y hardware) están anidados, tal como se ilustra en la figura 1. En cambio, tal como veremos más adelante, en computación neuronal estos niveles están distribuidos y no existe distinción entre hard y soft porque estructura y función coinciden. Veamos primero el caso usual para marcar después las diferencias y analogías en computación neuronal.



**Figura 1.** Niveles Anidados en Computación Simbólica.

Volviendo a la figura 1 y tal como comentábamos anteriormente, a nadie se le ocurre pasar directamente desde la formulación de un problema a nivel de conocimiento hasta la propuesta del circuito digital (combinacional y secuencial) que lo resuelve. Todo el edificio simbólico está basado en el «divide y vencerás». Primero, se plantea el problema usando lenguaje natural y el metalenguaje propio del dominio (Matemáticas, Economía, Ingeniería, etc.). Después se proponen estructuras de datos y algoritmos y, finalmente, se programa usando las primitivas de un lenguaje y un conjunto de facilidades de entorno que hacen más fácil la comunicación con la máquina. El nivel hardware se ignora. Podemos resumir las características de la computación simbólica en los siguientes puntos:

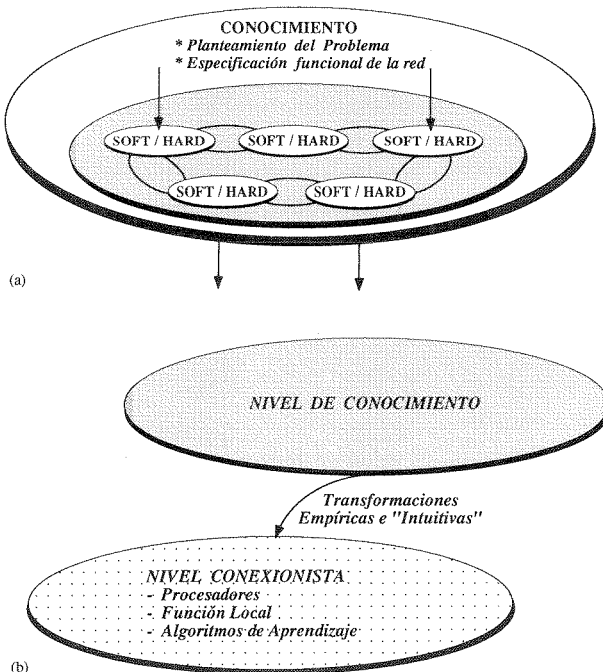
- S.1. Procesador único y «grande».
- S.2. Clara distinción entre hardware y software.
- S.3. Necesidad de programación externa en lenguajes de alto nivel.
- S.4. Codificación arbitraria, función del problema.
- S.5. Existencia de un conjunto de transformaciones (entornos, editores, ensambladores, sistemas operativos, ...) no accesibles a nivel de programador que trasladan la computación desde las primitivas del lenguaje hasta los procesadores.
- S.6. Hardware sin cambios.
- S.7. Dificultad en la programación del aprendizaje.
- S.8. Existencia de todo un cuerpo de conocimiento (por ejemplo, Curr. ACM).

Hemos comentado estas trivialidades para resaltar por comparación las carencias del estado actual de la computación neuronal.

La primera distinción entre conexionismo y computación simbólica es que no todos los problemas admiten una solución conexionista. Deben cumplirse las exigencias usuales en el paralelismo y en algunos aspectos de la lógica borrosa: vaguedad o incertidumbre en el conocimiento del problema, ser segmentable, modular y con necesidad de solución en tiempo real y pensado en medios cambiantes y poco accesibles que aconsejen la autoprogramación.

La segunda gran distinción entre el conexionismo y la computación simbólica es el paso directo desde el nivel de conocimiento al nivel de procesadores, sin el apoyo en los niveles intermedios (estructuras de datos, algoritmos, ensambladores, editores, sistemas operativos, entornos, etc.). Los esquemas de la figura 2 ilustran esta situación.

En la figura 2.a se muestra el equivalente conexionista de la figura 1. Ahora, del nivel de conocimiento en el que se plantea el problema y se dan unas especificaciones mínimas sobre la topología de la red, se pasa directamente al nivel de los procesadores, donde estructura y función coinciden, de forma que hardware y software son una misma cosa, salvo la parte del software que queda ahora distribuida sobre el valor de las conexiones.



**Figura 2.** Saltos de Nivel en Computación Neuronal. a) Paso del nivel de conocimiento al nivel hard/software que define la red. b) Ilustración del carácter empírico de la transformación.

Al no existir metodología, desarrollos teóricos y tecnologías intermedias, este paso se realiza mediante procesos empíricos de carácter intuitivo, tal como se ilustra en la figura 2.b. No es usual encontrar un texto sobre Redes Neuronales Artificiales en el que se nos enseñe a seleccionar arquitecturas, funciones locales y algoritmos de aprendizaje para familias de tareas genéricas.

Veamos ahora de forma más detallada lo que caracteriza a la computación neuronal, en comparación con la computación simbólica convencional.

- N.1. El modelo de máquina es tipo P-RAM con arquitecturas SIMD («Simple Instruction Multiple Data») por capas. Es decir, son máquinas RAM paralelo en las que dentro de cada capa la computación es homogénea. Todas las neuronas de una capa realizan el mismo tipo de computación genérica sobre datos diferentes, en general solapados. Son diferentes sin embargo los valores de los parámetros que especifican la computación local en cada coordenada como consecuencia del proceso de aprendizaje.
- N.2. El número de procesadores es muy elevado, pero la capacidad computacional de cada procesador es pequeña y, además, aparece directamente accesible al «programador» (diseñador de la red) en las etapas previas al comienzo del aprendizaje. Veremos después que existe un límite a la complejidad de la computación local consecuencia de las exigencias de autoprogramación.
- N.3. En la computación neuronal más genuina los procesadores deben ser autónomos, de forma que al tener que incluir la capacidad estructural necesaria para el aprendizaje, su arquitectura ya no es tan sencilla. La autoprogramación consume mucha más área de silicio que la computación. Esto limita el tamaño del «grano» cuando la computación paralelo pasa a ser neuronal, ya que para ser neuronal necesita poder ajustar parámetros y el ajuste consume más recursos que la propia computación. Así, una computación local de cierto grado de complejidad ya no podría ser neuronal.
- N.4. No existe distinción entre hard y soft. En computación neuronal, estructura y función coinciden.

Aunque se olvida con cierta frecuencia, toda computación neuronal es hardware. Toda simulación de redes neuronales en máquinas von Neumann, sólo tiene interés como etapa intermedia en el desarrollo de la red en la que se evalúan modelos, arquitecturas, algoritmos de aprendizaje y factores externos tales como el espacio de características y los conjuntos de entrenamiento en aprendizaje supervisado. Es decir, la computación neuronal está encaminada al cálculo paralelo a nivel hardware con parámetros ajustables en tiempo real, sin conexión alguna con el «host».

Ahora bien, la implementación depende del tipo de modelo (sumas, productos y sigmoide; max, mín y complemento; subconjunto de los operadores lógicos relacionales del C; etc.), de los algoritmos de aprendizaje (supervisado, no supervisado) y del grado de autonomía y plasticidad de los procesadores. Así, bajo el nombre de computación neuronal se incluyen procesos de computación distribuida y autoprogramable de grados de complejidad muy distintos.

- N.5. La codificación neuronal es parcialmente emergente, en contraposición al carácter arbitrario de la codificación en el caso de la computación simbólica, donde es el programador quien impone el código. Aquí, en las redes neuronales, lo único que es arbitrario es la asignación de significados a las variables de entrada y salida. El resto viene impuesto por la arquitectura de la red.
- N.6. Tolerancia a fallos. La computación neuronal es esencialmente segura porque tolera los fallos. De hecho, cada neurona artificial no se entera de la existencia de fallos más que a través del incremento del error. Este incremento lo trata el algoritmo de aprendizaje como a cualquier otro no procedente de fallos en la red.

Las redes neuronales son tolerantes a fallos porque son redundantes. El fallo deja de tolerarse cuando han desaparecido tantos procesadores que la red residual ya no es capaz de «copiar» con la misión de duplicar y generalizar el conjunto de pares  $(x_p, y_p)$  usados en el entrenamiento [Delgado, 1978; Delgado y Mira, 1979; McCulloch, 1960; Mira y Delgado, 1987; Mira y col., 1981, 1983].

- N.7. La Autoprogramación es parcial. No hay computación neuronal sin aprendizaje. La red es una estructura de cálculo que se autoprograma ajustando sus parámetros para minimizar una función de coste. Por consiguiente, cuando pasamos del nivel de conocimiento al nivel de los procesadores hacemos una partición del conocimiento inicial. Parte del mismo queda impreso en la arquitectura de la red, la conectividad, la función local y los mecanismos de aprendizaje. El resto se adquiere por entrenamiento.

Dicho de otra forma, la red es un procesador universal porque lo son sus neuronas y su esquema de conectividad. Esa estructura universal (genérica) la debe dar el «programador» (diseñador). El resto lo da el aprendizaje que proyecta ese procesador universal en uno específico, de acuerdo con el conjunto de valores de entrenamiento, que por supuesto también elige el programador.

- N.8. Inspiración en la Neurología. La penúltima propiedad (debería ser la primera) que caracteriza a la computación neuronal es su inspiración en la neurología, de donde toma el nombre y cierto «aura» de misterio y propiedades computacionales supuestamente análogas a las de los seres vivos.

Tras la propuesta inicial de Warren S. McCulloch y Walter Pitts, S.C. Kleene [1956] reformula las redes como representadoras de «sucesos regulares» y establece el origen de la teoría modular de autómatas finitos. En toda esta etapa inicial de las redes neuronales se camina muy cerca de la biología, buscando el soporte neuronal del reconocimiento de caracteres (problema de los «universales»), el control jerárquico y heterárquico, la autoorganización, la memoria asociativa o el cálculo seguro a partir de elementos sujetos a error [Kilmer y col., 1968; Kohonen, 1972; McCulloch, 1960; Nilsson, 1965; Shannon y McCarthy, 1956; Widrow, 1962; Yovits y Cameron, 1960].

Desafortunadamente, la mayor parte de las motivaciones biológicas iniciales se han perdido. En el renacimiento de las redes neuronales casi nadie discute el modelo de computación local aunque hay evidencia anatómica y fisiológica [Beach y col., 1960; Braitenberg, 1977; Bullock, 1979; Bullock, 1977; Hubel y Wiesel, 1977; Rockmann y Moraga, 1991; Schmitt y Worden, 1979] más que suficiente para proponer modelos más complejos (expansiones polinómicas y logarítmicas de los espacios de entrada y salida, retardos analógicos y digitales, condicionales, funcionalidades específicas, computación bioquímica en los canales de Na, K, Ca en membrana, etc.).

N.9. Falta de Metodología. Uno de los problemas fundamentales en Computación Neuronal es la falta de metodología, tanto de análisis como de síntesis. Hay un problema de análisis cuando, dada una red, queremos calcular la función que realiza de forma global, explicando el por qué de la configuración final. Inversamente, hay una situación de síntesis cuando, dado un problema, tenemos que proponer la arquitectura y las funciones locales de la red neuronal que lo resuelve. Para ninguno de estos tipos de problema existen procedimientos generales de solución. Sólo algunas «pistas», en el mejor de los casos [Caelli y col., 1993; Mira y col., 1993a; Riseman y Hanson, 1987; Rovaris, 1992].

Por consiguiente, no podemos hablar de redes en general, como la nueva versión del «General Problem Solver» de Newell, Shaw y Simon. Es necesario especificar la tarea y el dominio. Es decir, hay que hablar de redes específicas para familias de computaciones (clasificación, decisión, control, identificación, etc.) aunque para cada familia la red de nuevo es genérica.

Esta perspectiva exige reflexionar primero sobre las exigencias computacionales del problema y después sobre la adecuación y conveniencia de su solución neuronal. No todos los problemas tienen solución eficiente (y preferible a la simbólica) a nivel neuronal. Hace falta segmentabilidad de formulación paramétrica y no existencia de soluciones algorítmicas mejores. Hablaremos de nuevo de estos temas al final del trabajo.

Finalmente, admitida la posibilidad y conveniencia de la solución neuronal, todavía necesitamos procedimientos para seleccionar arquitecturas (capas, columnas y «barriles»), conectividad (total, local tipo inhibición lateral, centro-periferia, etc.), operadores universales y mecanismos de ajuste de los parámetros de los operadores. Todos estos procesos deben realizarse sin olvidar que nuestra meta final es conseguir una red de procesadores (hardware) que soporte físicamente a la red neuronal [Clarkson y Ng, 1993; Mira y col., 1994; Nordstrom y Suensson, 1992; Treleven y col., 1989; Vellasco y Treleven, 1993].

#### **4. PUENTE CONEXIONISTA-SIMBÓLICO**

Históricamente fue primero la computación neuronal. Después, (con algunos actores y autores repetidos y otros nuevos) aparece la computación simbólica y sus exponentes más conocidos: la Inteligencia Artificial (IA) inicial (búsqueda y heurística) y los Sistemas Basados en Conocimiento (SSBC) después. En ambos casos se buscan procedimientos para hacer computacional el conocimiento humano no analítico (el numérico ya lo era).

La percepción, la memoria asociativa, el aprendizaje considerado como acumulación de conocimiento representable, la planificación motora, la decisión y la creatividad se convirtieron en metas deseadas para los sistemas computacionales.

Ante las evidentes dificultades con las que se han encontrado la IA y los SSBC para resolver estas cuestiones, hacia la mitad de los años 80 se volvieron de nuevo los ojos hacia la computación neuronal [Rumelhart y col., 1986], pero en vez de buscar nuevos modelos de computación local, (junto con la inspiración que ofrecen los principios organizacionales y estructurales del sistema nervioso) se repiten en gran medida los errores del pasado. Esto ha llevado a un cierto agotamiento del modelo usual en computación neuronal.

Como intento de escape de esta situación de conflicto entre ambos paradigmas (simbólico y neuronal), es cada vez más frecuente encontrar propuestas de una formulación híbrida entre la computación simbólica y la neuronal, de forma que podamos quedarnos con lo mejor de ambas. Hay un primer tipo de propuestas que se ilustra en la figura 3. Podríamos llamarlas propuestas de aproximación superficial porque tanto la componente simbólica como la conexionista interactúan sobre máquinas von Neumann, la computación neuronal está simulada y las estructuras de datos son compartidas. Se introduce el procesamiento simbólico para la representación de las variables y se usa el aprendizaje conexionista donde hay imprecisión. De hecho, estas propuestas establecen la equivalencia entre CBL («Case Based Learning» o aprendizaje basado en casos) y el conjunto de entrenamiento de la red

y puede usarse para refinar bases de conocimiento a partir de ejemplos [Boticario y Mira, 1994; Dietterich y Flann, 1989; Mitchell y col., 1986; Shavlik y Towell, 1989].

En otras ocasiones se busca la computación híbrida para pasar por analogía directa desde EBC («Explanation Based Learning» o aprendizaje basado en explicaciones) a redes y se usa el aprendizaje en redes para refinar la teoría del dominio. En ambos casos, la formulación final es simbólica.

Frente a esta alternativa superficial existe otra, a la que podríamos llamar profunda, en la que se buscan Sistemas Expertos Conexionistas y Redes Neuronales Simbólicas (inferenciales), desde el nivel de representación. Hay, al menos, dos caminos posibles:

1. Extensión del conexionismo
2. Representación conexionista del edificio simbólico.

La extensión del conexionismo supone aumentar la capacidad computacional de las redes neuronales sin perder sus características más genuinas (arquitecturas paralelo, autonomía en los procesadores y capacidad estructural para soportar la autoprogramación). Para ello será necesario avanzar en varios frentes:

- \* Extensión de la computación local.
- \* Desarrollo de entornos (neurosimuladores).
- \* Desarrollo de arquitecturas genéricas.
- \* Avances teóricos y metodológicos.

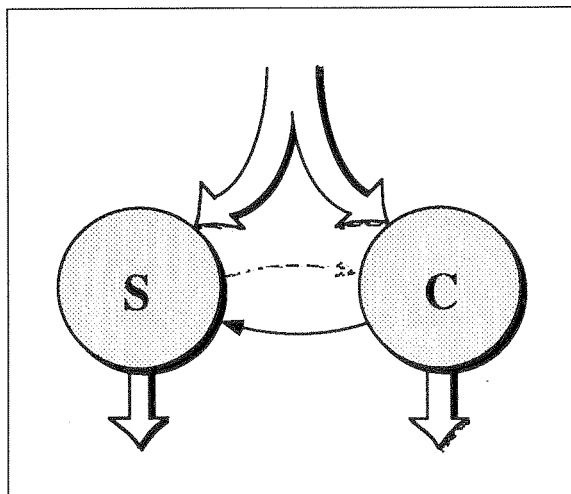


Figura 3. Conexión Superficial Simbólico-Conexionista.

De los cuatro aspectos, hablaremos sólo del primero (paso de la analítica a la inferencia local). Los otros tres caen fuera del alcance propuesto para este trabajo y tienen que ver con la falta de cuerpo propia de toda la computación neuronal.

Al igual que en toda Facultad de Informática parece lógico que existan asignaturas de Lógica, Electrónica Digital, Estructura y Tecnología de Computadores, Programación, Sistemas Operativos e Ingeniería del Software, por citar sólo algunos ejemplos, en una hipotética Facultad de Informática Conexionista no existiría nada entre el planteamiento del problema y la red de procesadores autoprogramables que lo resuelve.

Este es el mayor escollo para la extensión del conexionismo. No hay nada parecido a una «Lógica Modular con capacidad de autoprogramación», «Electrónica Neuronal», «Arquitectura de Redes Neuronales», «Preprogramación Conexionista para tareas genéricas», o «Ingeniería del software neuronal». Tampoco existen programas traductores tales que, dado un cierto nivel de especificación funcional de una red, nos genere de forma automática la red compilada.

Volvamos ahora al problema de extensión de la función local en el modelo analógico en dos pasos:

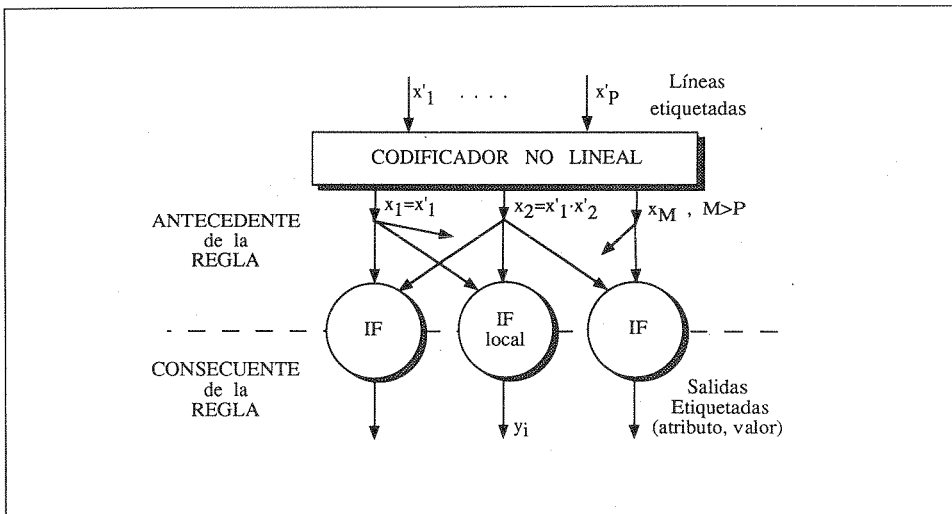
1. Expansión no lineal de los espacios de entrada y salida sobre líneas etiquetadas.
2. Sustitución de la función local por un condicional.

La primera extensión supone mantener la suma ponderada como función local pero extendiendo el espacio de entradas mediante codificadores no lineales (polinómicos, exponenciales) que representan la información en pares *<atributo,valor>*. Además, se incluye el tiempo como variable de cálculo al permitir que las zonas de muestreo estén sobre memorias FIFO («First In First Out») [Delgado y col., 1989; Mira y col., 1983, 1991a, 1991b, 1992, 1993c, 1994].

La propuesta del segundo paso de expansión trae más consecuencias. Consiste en sustituir la función local (suma más sigmoide) por un condicional tal como se ilustra en la figura 4. La inclusión como caso particular de la formulación analítica es evidente. El antecedente de la regla evalúa la condición de disparo y el consecuente (IF múltiple) genera la decisión, que es una nueva línea etiquetada [Mira, 1989; Mira y Delgado, 1988, 1991b]. Hemos presentado la generalización para el modelo unidimensional y sin realimentación para no complicar el esquema. Es evidente sin embargo que no hay pérdida de generalidad.

Esta extensión elimina las restricciones del operador suma, sustituye la decisión no lineal por una LUT («Look-Up-Table»), incluye las redes borrosas [Mira y col., 1993b] como caso particular (función de pertenencia en la LUT) y posee la

complitud necesaria en toda representación. Es decir, al igual que las puertas NAND (o las NOR) son operadores completos en electrónica digital, de forma que cualquier otra función puede sintetizarse con el uso exclusivo y repetido de este tipo de operador, el IF es completo como estructura de control [Böhm y Jacopini, 1966]. El punto que quizás no haya quedado claro y que es crucial es el carácter hardware del condicional y por consiguiente la complejidad media necesaria para su implementación analógica o digital [Sánchez-Sinecio y Newcomb, 1993; Vidal, 1988; Wawrzynek y col., 1993; XILINX, 1993].



**Figura 4.** Extensión de la computación local que permite pasar de la analítica a la algorítmica. Cada neurona se sustituye por un condicional.

Es decir, el puente conexionista-simbólico se establece al sustituir las redes de neuronas artificiales por redes de condicionales hardware que evalúan en paralelo y en tiempo real la satisfacción o no satisfacción de una condición genérica espacio-temporal sobre los datos que cubre su campo receptivo (máscara sobre memorias FIFO). Por eso decíamos al comienzo que el modelo de computación implícito en las redes neuronales es el correspondiente a las máquinas RAM paralelo (P-RAM) tipo SIMD por capas (única-instrucción-dato-múltiple). Esa única instrucción es el IF múltiple, tal como se ilustra en la figura 5.

Como el IF tiene que autoprogramarse, las restricciones ahora hacen referencia a la variedad de operadores admitidos para el campo de condición. Es decir, ¿cuál es el operador universal que permite su modificación de acuerdo con la contribución al error global producido por la red?. Recordemos que no hay

computación neuronal sin aprendizaje y las reglas del aprendizaje (supervisado o no) son los algoritmos de modificación de los parámetros de computación local (operadores del IF en este caso) de acuerdo con criterios globales de evaluación del comportamiento de la red.

¿Cómo son entonces las Redes Inferenciales (o los Sistemas Expertos Conexionistas)? Son redes como las que se ilustran en la figura 6, que mantienen la anatomía de las redes analíticas, sus esquemas de realimentación y conectividad local y el carácter propio de la computación paralelo por capas y secuencial entre capas.

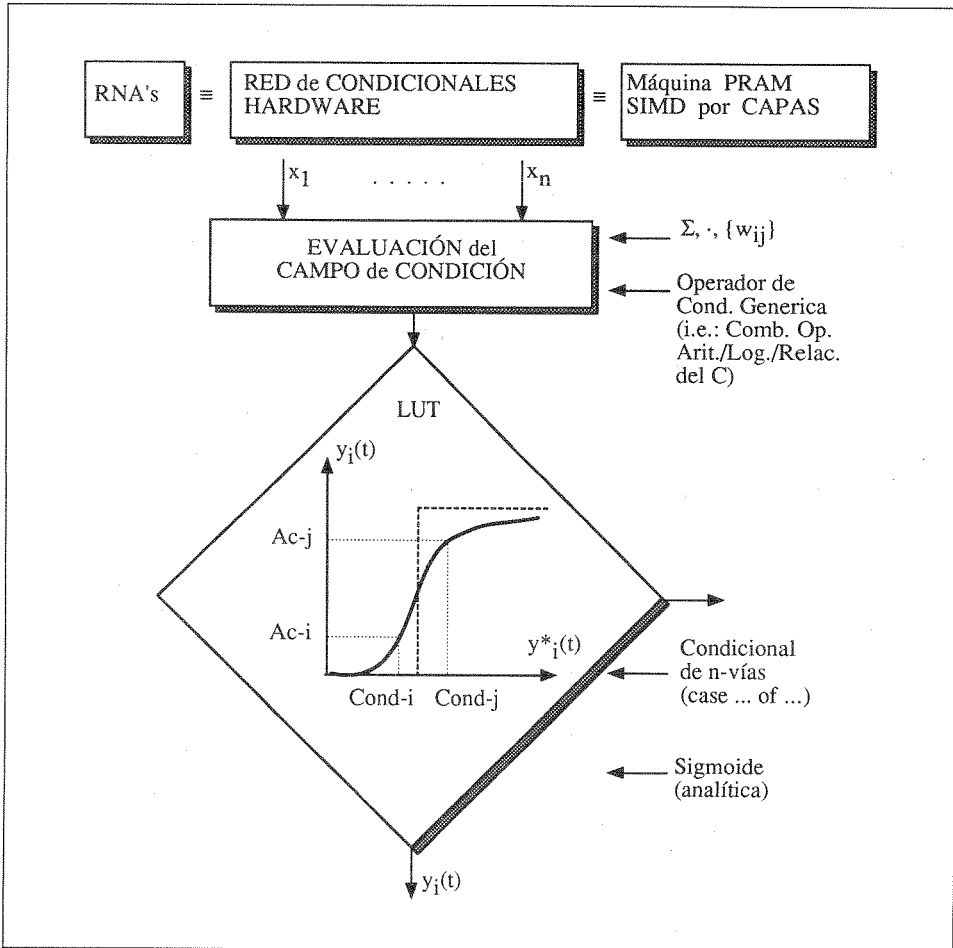
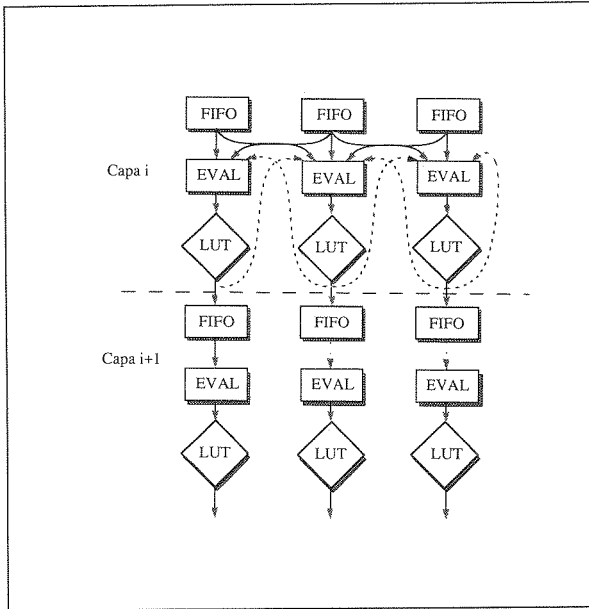


Figura 5. Resumen de la propuesta de Red de Neuronas Artificiales como red de condicionales hardware.



**Figura 6.** Redes Neuronales Inferenciales (o Sistemas Expertos Conexionistas).

Aparte del hecho evidente del carácter «hardware» de las redes inferenciales, podemos resumir un conjunto de puntos que marcan las analogías y diferencias con los SSBC convencionales.

1. Cada neurona inferencial es un módulo de conocimiento genérico junto con su mecanismo de aplicación. Un «agente no reductible» capaz de participar de forma cooperativa-competitiva en todas aquellas tareas que se puedan paralelizar. Es decir, la red infiere de acuerdo con el paradigma usual en la IA distribuida con las fuertes condiciones de contorno de la autoprogramación e implementación hardware.
2. La función global es la que emerge de una red de condicionales con conectividad local. La evaluación y la LUT locales pasan de genéricas a específicas por aprendizaje. La base de conocimiento (BC) y el sistema de inferencia están juntos en cada neurona y distribuidos en la red, de forma que la BC global es la propia red.
3. La inferencia es refleja. La red «no sabe» que está razonando. Cuando se acumulan determinadas configuraciones espacio-temporales de datos y hay sincronismo, los campos de condición disparan sus LUTs. La respuesta de la red depende de la arquitectura y de los datos y está disponible siempre en las LUTs de las unidades accesibles.

4. Tenemos así computación simbólica sobre arquitecturas neuronales. El coste es que los nuevos procesadores (IFs hardware automodificables) son más complejos que los NAND y los sumadores. Su nivel de complejidad es análogo al de un microprocesador.
5. No es sencillo posibilitar estructuralmente y realizar de hecho la autoprogramación de estas redes, no hay metodología y no son adecuadas para todo tipo de función.

El consuelo es que tampoco es sencillo programar sistemas expertos en lenguaje ensamblador y ese es el equivalente en computación neuronal. Pero cuando la red está programada, las diferencias simbólico-conexionista sólo aparecen en el dominio del observador externo, nunca en el dominio propio de los procesadores, sean estos NAND, sumas o IPs<sup>1</sup>.

La propuesta de representación conexionista del edificio simbólico queda fuera del alcance de este trabajo. Sólo queremos esbozarla para dar una visión global del problema, poniendo de manifiesto que las perspectivas conexionista y simbólica son complementarias a nivel de desarrollo de aplicaciones y equivalentes a nivel de procesadores.

El procedimiento que se sugiere es seguir la metodología usual en el desarrollo de SSBCs (Adquisición del Conocimiento, Representación, Inferencia, Programación y Entornos, Aprendizaje (EBC, CBL, ...), Evaluación y Refinamiento).

La idea de partida es que adquirir conocimiento sobre una tarea es equivalente a modelar esa tarea mediante una red, distinguiendo las entidades y la estructura inicial (esqueleto) y proporcionando un mecanismo de construcción incremental, ajuste de parámetros y poda selectiva.

La función global siempre admite la estructura genérica de un proceso de clasificación lo que, aunque pueda parecer lo contrario, no es nada restrictivo. El diagnóstico, la decisión, el tratamiento digital de señales, el reconocimiento de caracteres, la visión artificial, la planificación y el control son tareas que admiten una formulación tipo clasificador.

La representación lógica y por reglas de producción hemos visto que encuentra su contrapartida inmediata en redes con la limitación de la conectividad. Sólo se pueden enlazar las reglas que están conectadas a nivel hardware.

---

<sup>1</sup> Toda la redacción de este apartado referente al puente simbólico-conexionista procede esencialmente del trabajo de J. Mira «De la Computación Neuronal a la Computación Simbólica», «Journal CETTICO», N. 10, junio, (1994), 21-31.

Finalmente, el intento de reformulación conexionista del aprendizaje simbólico pasa por la identificación de un conjunto de primitivas comunes a los procesos de inducción, deducción y abducción. La identificación, indexación, comparación y adaptación son tareas para las que existe una arquitectura de redes neuronales genéricas que las implementa. Su desarrollo es objeto de trabajo actual en el grupo.

## **5. BASES ANATOMO-FISIOLÓGICAS DE LA COMPUTACIÓN NEURONAL**

Parece inicialmente tautología hablar de bases biológicas de la computación neuronal puesto que no existen más neuronas que las biológicas. Lo que actualmente se conoce como redes de neuronas artificiales corresponde a un modo de computación analógica no lineal, distribuida, organizada por capas y autoprogramable que utiliza como computación local un modelo muy primitivo de neurona biológica. El argumento de este apartado del trabajo es señalar la gran complejidad de la neurona real y la posibilidad de obtener modelos computacionales mucho más eficientes que el del sumador. Esta perspectiva creemos que es ventajosa al menos en cuatro sentidos:

1. No volver a reinventar la rueda.
2. Acercarnos a la Neurociencia en su propósito de comprender el funcionamiento de los sistemas nerviosos (modelado, codificación, circuitos locales, plasticidad, desarrollo y evolución).
3. Obtener nuevos modelos computacionales (analíticos no lineales, recurrentes, inferenciales y basados en conocimiento) más ricos y eficientes que los actuales cuando se usan en aplicaciones de identificación, control, predicción, percepción artificial y robótica.
4. Obtener inspiración sobre nuevas arquitecturas incrementales y distintos estilos de computación basados en la morfogénesis, la genética, la programación factorial, la poda selectiva y la biología molecular.

Estamos convencidos que en la neurociencia existe inspiración no sólo para los nuevos modelos de computación local sino también para la búsqueda de nuevas arquitecturas, principios organizacionales (cooperatividad, autopoiesis, optimización evolutiva) y nuevos mecanismos de aprendizaje más allá del refuerzo y la asociación [Braitenberg, 1977; Bullock, 1979; Bullock y col., 1977; De Felipe y Fariñas, 1992; Hebb, 1949; Hubel y Wiesel, 1962, 1969; Mountcastle, 1979; Schmitt y Worden, 1979].

El final de este apartado va a ser la propuesta de un conjunto de exigencias computacionales mínimas para los modelos de neuronas y redes neuronales que pretendan parecerse a la biología. Es evidente que la computación artificial no podrá duplicar a corto plazo a la computación neuronal. Dicho de otro modo, todas las

características conocidas en neurociencia no pueden ser incorporadas de forma inmediata a las redes de neuronas artificiales (RNAs). Entre otras razones por la diferencia clara entre los cristales semiconductores (Si, AsGa, ...) y el tejido biológico del que están hechos los procesadores neuronales. Lo mismo le ha ocurrido a la IA frente a los procesos cognitivos.

Sin embargo, existen otras características (no linealidad, retardo gradual, realimentación y poda selectiva, especialización funcional, distinción entre maduración y aprendizaje, aceptación de la existencia de distintos estilos de computación, arquitecturas incrementales, etc.) que sí que son susceptibles de síntesis artificial a corto plazo. Lo único que exigen es un cierto cambio de mentalidad para aceptar que la complejidad de una neurona artificial debe ser al menos del orden de la correspondiente a un microprocesador de última generación junto con las facilidades estructurales que posibiliten su auto-programación para distintas familias de funciones.

Organizaremos este apartado en las siguientes secciones:

1. Arquitecturas modulares.
2. Ruta de datos (conectividad).
3. Computación local.
4. Módulos funcionales a nivel intermedio.
5. Funciones de memoria.
6. Generadores de patrones de respuesta.

### **5.1. Arquitecturas Modulares**

Desde Cajal parece evidente que todos los sistemas nerviosos (SN) poseen una estructura modular con muchos procesadores (más de  $10^{11}$ ) de una gran variedad de tipos distinguibles a nivel molecular y anatomo-fisiológico, organizados en grupos funcionales repetidos en los individuos de una especie y con homólogos en todas las especies [Bullock y col., 1977].

Admitida la modularidad, parece razonable elegir un nivel de descripción y estudiar las arquitecturas correspondientes a ese nivel en términos de lo que se considera «módulo elemental» del nivel y de sus esquemas de conectividad próxima y distal en las distintas fases de desarrollo. A menudo se olvida (en el campo de las RNAs), que las neuronas «adultas» y las arquitecturas de los SNs «maduros» son el resultado de una secuencia de procesos genético-evolutivos de gran complejidad de los que podemos obtener inspiración para la síntesis de RNAs con arquitecturas plásticas. En estos procesos ya va implícito un primer nivel de aprendizaje. No hay que olvidar sin embargo que todo lo que hay de «genéticamente determinado» en las neuronas biológicas, tenemos que inyectarlo en la arquitectura inicial de las RNAs.

El resto se superpone a esta estructura inicial a través del aprendizaje [Mira y col., 1994].

Sin ánimo de ser exhaustivos, y sólo para ilustrar la riqueza del campo biológico como fuente de inspiración en el diseño de RNAs podemos distinguir, al menos, los siguientes puntos referentes a las arquitecturas neuronales [Bullock y col., 1977; Mountcastle, 1979; Schmitt y Worden, 1979; Segev, 1992].

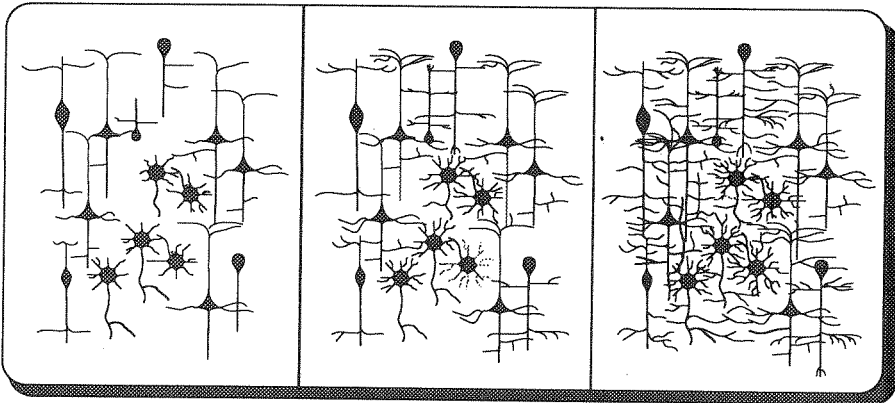
- A.1. Programación genética que controla, al menos parcialmente, la proliferación y diferenciación de las células nerviosas en distintos tipos y guía la conectividad y el desarrollo. Desafortunadamente, no es fácil transportar la genética al campo de las RNAs. Los trabajos iniciales sobre programas programadores y autómatas que se auto-reproducen (A. Burks y J. von Neumann), no han tenido el desarrollo deseado. Bajo el nombre de algoritmos «genéticos» y «vida artificial» también existen algunos desarrollos que rozan la genética, pero muy de lejos.
- A.2. Morfogénesis (desarrollo, muerte selectiva y especificidad de conexiones). En la formación de las arquitecturas neuronales se presentan un conjunto de procesos con proliferación masiva de conexiones, migración, diferenciación y muerte selectiva por reabsorción de parte de esas conexiones y desaparición de las células que no encuentran su «diana» postsináptica.

La arquitectura neuronal nace con todas las células que van a sobrevivir y muchas más (por ejemplo, todos los procesadores locales de la red madura). Sin embargo en seis meses dobla el peso de la red y en cuatro años lo vuelve a doblar. Es evidente entonces que la maduración de la arquitectura ha consistido en aumentar de forma brusca la conectividad local con el crecimiento ordenado de dendritas y axones y con la reabsorción (eliminación) de las conexiones que no se han mostrado relevantes porque las aferentes correspondientes no eran significativas en el medio en el que se ha desarrollado la arquitectura.

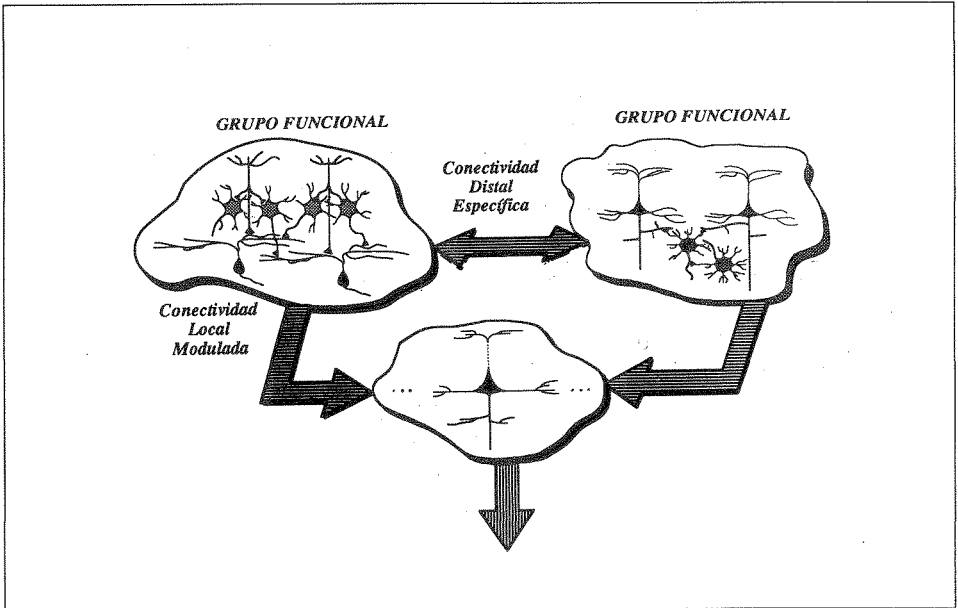
Las células postsinápticas son de vital importancia para mantener vivas a las neuronas presinápticas. Si una neurona presináptica no alcanza la postsináptica adecuada, puede degenerarse y desaparecer. Esta muerte selectiva durante el desarrollo, junto con la producción excesiva de conexiones y posterior poda selectiva, nos obliga a distinguir entre maduración de arquitecturas y posterior refinamiento por aprendizaje.

La figura 7 ilustra el crecimiento de la conectividad durante el desarrollo de la arquitectura. La posterior reducción tiene que ver con el tipo de información aferente durante el intervalo de desarrollo. Lo que nos interesa resaltar en este apartado es la riqueza de la morfogénesis y los procesos de maduración a la hora de definir arquitecturas incrementales de conectividad variable en el campo de las RNAs. Primero está la maduración y después el aprendizaje.

- A.3. Grupos Funcionales. La especialización en la computación neuronal es jerárquica y aparece a muchos niveles: órganos, grupos funcionales, tipos de neurona, etc. La figura 8 ilustra de forma esquemática la organización por grupos funcionales a los que por comodidad se suele representar en forma de capas. Decimos entonces que las arquitecturas neuronales son arquitecturas multicapa [Delgado, 1978; McCulloch, 1965; Moreno-Diaz y Rubio, 1979]. La retina y la corteza cerebral son ejemplos claros en este sentido, pero no debe olvidarse que la idea de base es el «grupo funcional», donde la computación parece ser homogénea y la conectividad es local y función de la distancia (está modulada), a diferencia de la conectividad específica, distal y no modulada entre los distintos grupos funcionales que integran una arquitectura. Veremos más adelante la conveniencia de agrupar las neuronas en distintos tipos (especialización funcional) y distinguir también distintos niveles y estilos de computación.
- A.4. Especialización Funcional. Las neuronas adultas aparecen como células especializadas en la adquisición, integración, transformación, generación, transporte y distribución de estados de excitación que se manifiestan a nivel eléctrico como potenciales lentos o potenciales de acción en forma de pulsos que se repiten con una frecuencia proporcional al nivel de excitación que supera un cierto valor umbral. Existen además otros tipos de manifestaciones de la función neuronal a nivel bioquímico molecular y electrónico.



**Figura 7.** Ilustración anecdótica del crecimiento de la conectividad durante el desarrollo. Al crecer la arborización dendrítica se expande de forma brutal el espacio de entradas de cada neurona. Después hay una reducción del espacio de entradas por reabsorción de ramas. Esta poda selectiva, junto con la muerte neuronal es la base de la maduración de las arquitecturas neuronales.

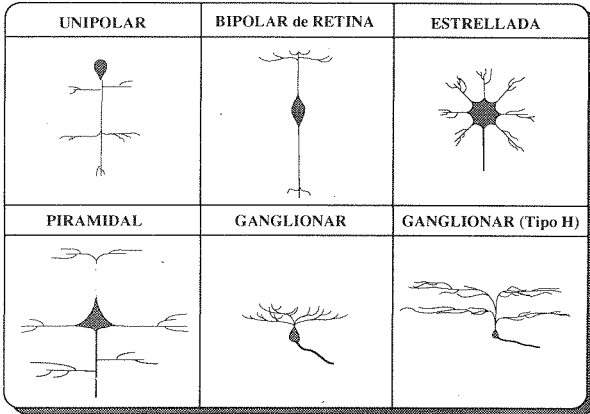


**Figura 8.** Arquitecturas neuronales organizadas en grupos funcionales. En cada grupo cooperan distintos tipos de neuronas, pero, probablemente, la computación es homogénea (ley de acción de masas de Lashley y gradientes funcionales de J. Gonzalo y Luria) y la conectividad es recursiva. La conectividad distal, entre grupos, es específica y menos tolerante a fallos.

La evidente especialización funcional de las neuronas se manifiesta esencialmente a través de su anatomía (forma de los campos dendríticos y axónicos, tipo de cuerpo celular, ...) y por su posición contextual en el SN (transductores, células integradoras aferentes, interneuronas, piramidales de corteza, motoras, etc.). Desde nuestra perspectiva computacional podemos distinguir al menos los siguientes tipos:

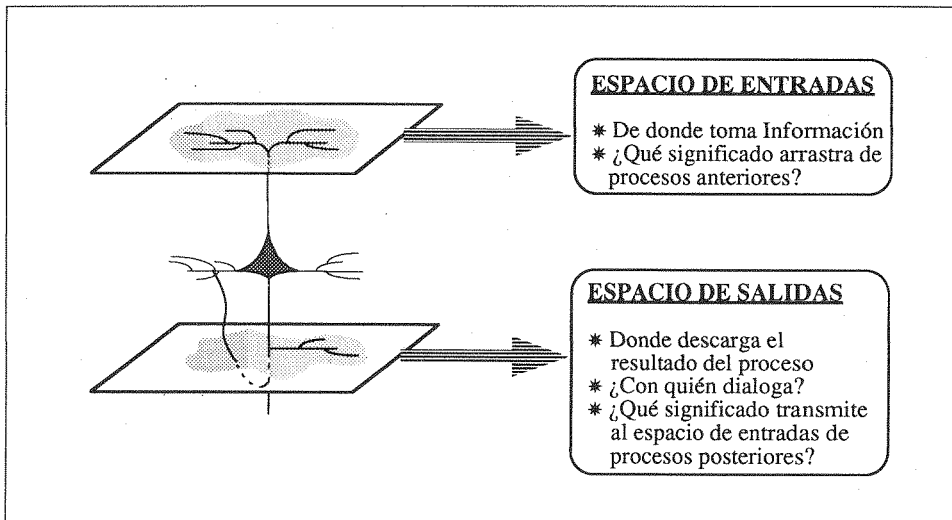
- \* Receptores (por ejemplo, conos y bastones).
- \* Detectores de propiedades espacio-temporales (por ejemplo células ganglionares en retina).
- \* Células de asociación (por ejemplo, interneuronas, amacrinas en retina, células de axón corto, estrelladas en corteza, ...).
- \* Células de integración (por ejemplo, piramidales de corteza).

La figura 9 ilustra algunos de estos tipos (unipolares, bipolares, estrelladas, ganglionares y piramidales), poniendo de manifiesto la región de la que toman información (campo dendrítico) y la zona de descarga (campo axónico). Es decir, su espacio de entradas y su espacio de salidas.



**Figura 9.** Tipos de neuronas a nivel anatómico.

La figura 10 muestra de forma esquemática la definición anatómica de estos espacios y su significado computacional. Para entender este último, no nos basta con conocer la conectividad y la función local. Necesitamos conocer también el contexto, es decir, donde se encuentra esa neurona y que preproceso ha experimentado la información antes de entrar en su campo dendrítico. La constante recodificación de la información aferente es clave para entender el significado computacional de cualquier proceso, tras la primera etapa de receptores.



**Figura 10.** Espacios de entrada y salida de una neurona. El contexto define el preproceso y su significado y transporta a la salida la acción de filtrado que ha introducido la computación local. El significado inicial más la transformación local dan lugar al nuevo significado de las señales del espacio de salidas, que será espacio de entradas para otras unidades o para las mismas, si hay realimentación por colaterales o a través de interneuronas.

Podemos tomar el camino visual como ejemplo para ilustrar la especialización funcional. La información va desde el receptor (retina) hasta las áreas de asociación pluri sensorial pasando por el cuerpo geniculado externo (tálamo), las áreas primarias y secundarias y los núcleos talámicos asociativos, tal como se ilustra en la figura 11.

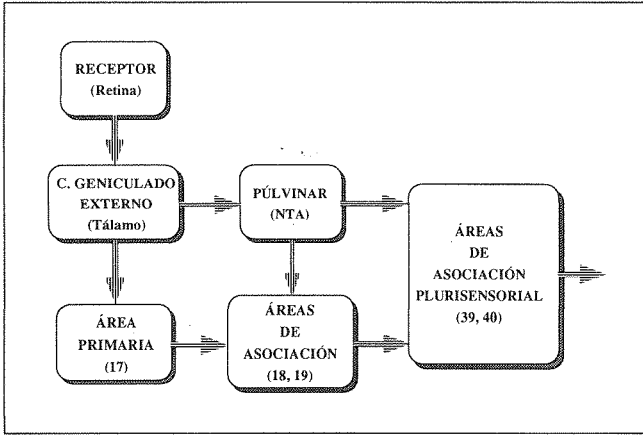
Centrémonos ahora en la retina. Un diagrama esquemático común a muchas retinas (rana, gato, paloma, ...) es el que se muestra en la figura 12. Está organizada por capas y consta de una primera capa de fotorreceptores, una capa plexiforme externa (CPE), la capa de bipolares, una capa plexiforme interna (CPI) y, finalmente, una capa de ganglionares, cuyos axones forman el nervio óptico [Bullock, 1979; Moreno-Díaz y Rubio, 1979; Schmitt y Worden, 1979; Wechsler, 1992].

Los fotorreceptores son transductores logarítmicos que convierten la luz en señal electrofisiológica siguiendo la ley de Weber-Fechner que describe la relación entre la intensidad del estímulo y la intensidad de la sensación experimentada por el sujeto. El número de fotorreceptores en la retina humana es del orden de  $130 \times 10^6$ , lo que nos da una medida de la frecuencia espacial de muestreo de la imagen a partir de la cual hay que recuperar la información.

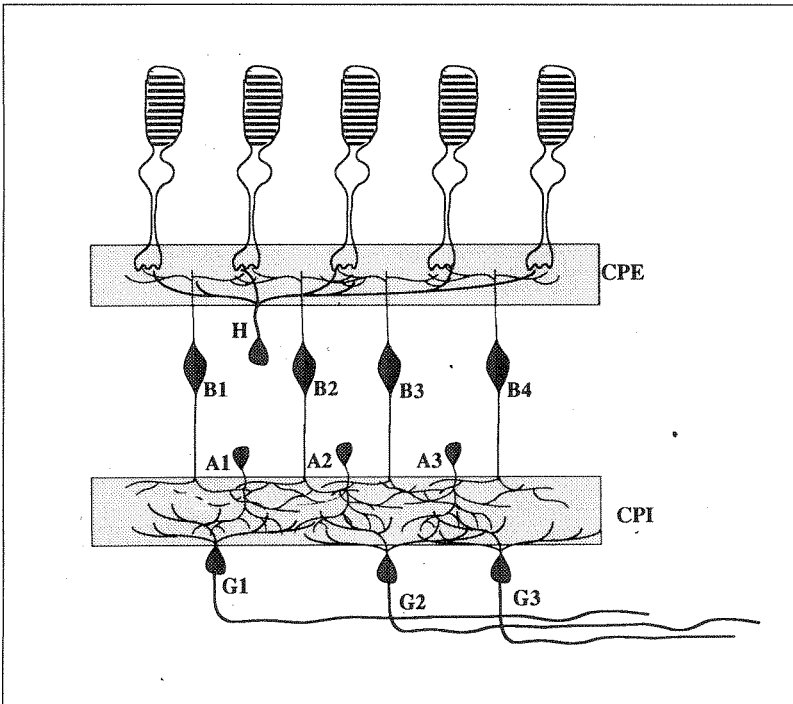
En la CPI, los terminales de los receptores hacen contacto con las células horizontales y las dendritas de las bipolares, formando un primer plexo de interacción horizontal (lateral) donde la función de las células horizontales sería de asociación, de forma que una célula bipolar pasaría a la siguiente capa plexiforme el resultado de su proceso directo de la información local que toma de los receptores más la parte correspondiente a la información aferente sobre otros receptores transportada y transformada por las horizontales. La existencia de sinapsis excitadoras e inhibitoras a este nivel, junto con los retardos permite suponer que a la salida de la CPI ya tenemos un primer preproceso con la extracción de características diferenciales espacio-temporales (contrastes, movimiento y posible direccionalidad).

Viene a continuación la capa plexiforme interna (CPI), formada por los pies de las bipolares, las amacrinas y los campos dendríticos de las ganglionares. Hay mayor densidad de contactos que en la CPE y casi siempre se observan díadas postsinápticas soportando de nuevo interacciones laterales más locales que las observadas en la CPE.

Las células ganglionares son unidades de integración y de detección de características espacio-temporales de los estímulos. Hay muchos tipos morfológicos de ganglionares pero desde el punto de vista computacional es más importante su clasificación en términos de la forma espacio-temporal de su campo receptivo. Además, a este nivel existen datos experimentales que nos permiten caracterizar con precisión la forma del campo, a partir de las características de los estímulos a las que se responde con mayor intensidad.



**Figura 11.** Soporte anatómico del camino visual. Incluye los grandes grupos funcionales por los que pasa la información desde los receptores hasta la corteza asociativa. De las áreas plurisensoriales la información pasa a otras zonas de corteza. Esta organización (Receptor, Núcleos, Áreas Primarias y Áreas de Asociación) se repite para todas las modalidades sensoriales.



**Figura 12.** Organización de la retina por grupos funcionales (capas): receptores, capa plexiforme externa (CPE) (con las horizontales (H) y sus contactos con pies de fotorreceptores y bipolares), bipolares (B1, B2, B3) y amacrinas (A1, A2, A3) de la capa plexiforme interna (CPI), de la que toman información las ganglionares (G1, G2, G3). Sus axones constituyen el nervio óptico que sube hacia tálamo y corteza.

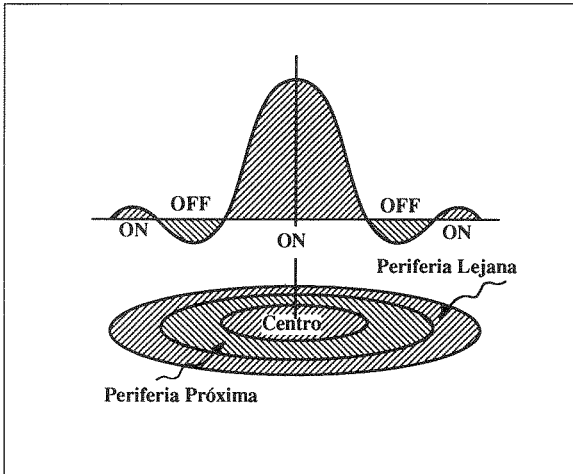
El área de la retina tal que al estimularla responde una ganglionar determinada, constituye el campo receptivo de esa ganglionar. Este campo receptivo estará formado en principio por todos los fotorreceptores que se encuentran dentro de la arborización dendrítica de esa ganglionar (campo receptivo anatómico). Sin embargo, debido a la existencia de interacciones laterales en las capas plexiformes (CPI y CPE), la ganglionar también responde al estimular zonas de la retina alejadas del área de fotorreceptores que cubre su campo dendrítico dando lugar a un campo receptivo funcional (CRF) mucho más amplio que el anatómico.

El CRF define el volumen de muestreo, del que se toma la información que procesa una ganglionar y define su computación. Los CRF están organizados en zonas alternadas de excitación e inhibición, tal como se ilustra en la figura 13. La organización dominante es del tipo «centro, periferia próxima, periferia lejana» y define las regiones de las que se toma información no sólo en (x,t), sino en (x,y,z), siendo  $z=f(x,y;t)$ . Dentro de cada región se admite que las sinapsis son predominantemente excitadoras o inhibitoras de forma que en una formulación analógica los «pesos» que definen el núcleo o la función ponderatriz son alternativamente positivos o negativos y modulados por la distancia. La forma analítica más usual es la diferencia de gaussianas,

$$w_{ij} = (\text{excitación en } j) (K_1 e^{-K_2 d_{ij}^2} - K_3 e^{-K_4 d_{ij}^2})$$

La organización de los CRFs es más compleja en animales inferiores que en los superiores, donde se traslada parte del proceso a corteza. En todos los casos podemos encontrar siempre las organizaciones centro-periferia tipo ON, OFF y ON-OFF, de distinto tamaño, simétricas y asimétricas, con selectividad direccional y con capacidad de aprendizaje de forma que la evolución temporal de los CRF en los distintos grupos funcionales determina la computación realizada en el camino visual. Obsérvese que al introducir el retardo temporal, el CRF muestrea sobre una FIFO en la que existen varias versiones retardadas de una imagen y de su preproceso. Sobre estas muestras se realiza ahora el resto de la computación, de acuerdo con las operaciones que comentaremos más adelante al hablar de la función local. El trabajo de R. Moreno y E. Rubio [1979] muestra la capacidad computacional y predictiva de un modelo general de retina basado en CRFs tridimensionales.

A nivel biológico, todavía sigue teniendo un valor paradigmático el trabajo de Lettvin, Maturana, McCulloch y Pitts [Lettvin y col., 1959] sobre la codificación del espacio visual de la rana *pipiens* mediante cuatro tipos de células ganglionares cuyos CRFs están sintonizados a estímulos importantes para la supervivencia.



**Figura 13.** Campos receptivos con áreas alternadas excitadoras e inhibitorias. La generalización al caso tridimensional permite describir gran parte de los resultados conocidos para distintas retinas. Tendríamos entonces volúmenes concéntricos alternativamente excitadores e inhibidores, con o sin simetría, con direcciones preferentes y, finalmente, estáticos o adaptativos. Esta última posibilidad permite entender un cierto tipo de aprendizaje como transformación del volumen del que se toman los datos.

A.5. Organización topográfica y topologías específicas. Un aspecto esencial en las arquitecturas neuronales es que nada parece arbitrario [Braitenberg, 1977; De Felipe y Fariñas, 1992]. Al coincidir estructura y función el estilo de computación se aleja del usual en los campos numérico y simbólico donde la codificación es arbitraria. En las redes neuronales la codificación es por líneas etiquetadas y por la propia topología de la red junto con la posible existencia de «marcadores fisiológicos» [Bullock, 1979; Bullock y col., 1977; Calvin y Graubard, 1979; Koch y Poggio, 1985].

El ejemplo más evidente de este orden es la organización topográfica de las vías sensoriales. Las distintas regiones del espacio receptivo de cada una de las modalidades sensoriales están representadas en corteza de forma ordenada. Es decir, las relaciones de vecindad en la periferia se preservan en las neuronas del SNC. Lo adyacente en el campo visual a nivel de retina se proyecta sobre neuronas adyacentes del córtex visual (retinotopía).

Existen además organizaciones específicas en columnas, estructuras tipo quiasma que duplican y cruzan la información, conexiones sensoriomotoras, etc., que van en contra de la estrategia usual en RNAs de conexión «todos contra todos» sin organización local-distal ni orden topográfico, salvo en los desarrollos específicos sobre memoria asociativa y redes autoorganizativas en las que se hace énfasis justo en estos aspectos de organización topográfica, como base de la codificación de la información aferente [Hubel y Wiesel, 1962, 1969, 1977; Mountcastle, 1979; Schmitt y Worden, 1979; Steriade y Llinás, 1988; Werner y Whitsel, 1973].

Lo que nos sugiere la biología es que no sólo en el caso de las redes autoorganizativas, sino siempre que vayamos a diseñar una red artificial, debemos partir de una arquitectura inicial que resuma todo nuestro conocimiento del problema, definiendo el «esqueleto» global de los distintos grupos funcionales, las conexiones distales entre estos grupos, su arquitectura y los distintos tipos de neuronas, los campos receptivos dentro de cada grupo funcional, etc.

A.6. Niveles y estilos de computación. Hasta ahora hemos supuesto que la neurona era el «átomo» en la arquitectura modular de SN. Está claro que esto es cierto pero existe tal riqueza de conexiones dendro-dendríticas y tal cantidad de procesos a nivel bioquímico que parece esencial distinguir otros niveles y estilos de computación [Calvin y Graubard, 1979; Koch y Poggio, 1985; Segev, 1992; Steriade y Llinás, 1988], aparte del neuronal. Sin ánimo de ser exhaustivo podríamos distinguir los siguientes niveles, en orden de integración creciente:

- \* Bioquímico molecular.
- \* Microcomputación analógica.
  - Circuitos dendro-dendríticos.
  - Computación sináptica (díadas, tríadas).
- \* Redes que enlazan neuronas.
  - Retina.
  - Corteza.
  - Núcleos Talámicos.
- \* Arquitecturas de grupos funcionales (redes de redes).
  - Integración plurisensorial.
  - Conexiones corteza-corteza y corteza con núcleos talámicos.
  - Conexiones sensorio-motoras.
- \* Circuitos masivos de computación cooperativa (ley de acción de masas de Lashley, códigos estadísticos de Beurle, computación factorial de Luria y gradientes funcionales de J. Gonzalo) [Beach y col., 1960; Delgado y col., 1989; Gonzalo, 1952; Luria, 1974; Mira y col., 1980].

La idea esencial en la teoría de niveles es que todos ellos coexisten. Dicho de otro modo, el concepto de nivel es una argucia que hemos inventado para aclararnos nosotros en la comprensión de una fenomenología que nos desborda porque la neurona es algo muy complejo como para ser considerada un «procesador elemental». De hecho su comportamiento se parece más a un gran centro de integración de información con 60.000 procesadores; tantos como sinapsis.

No podemos extendernos más en este apartado. Sólo queremos señalar la riqueza de las arquitecturas neuronales y la coexistencia de distintos estilos de computación, al menos tantos como niveles. Así, a nivel neuronal no sólo se procesa

información de acuerdo con los modelos analógico y digital, sino que existe procesamiento algorítmico y, posiblemente otros muchos diferentes a los usuales en las ciencias de la computación artificial. De todos ellos, los más significativos son los asociados a la bioquímica molecular.

A.7. Autonomía. Parece de nuevo innecesario recordar que las neuronas biológicas son autónomas. Es decir, no hay nada fuera de la neurona que contribuya a su computación y a su aprendizaje. Ya comentamos en la primera parte de este trabajo las consecuencias que arrastra la autonomía a la hora de establecer el nivel mínimo de complejidad de la implementación. La proporción entre el área de silicio dedicada a la programación frente a la correspondiente al cálculo local que se programa es del orden de 10 a 1. Por eso en RNAs el cálculo local no puede ser muy complicado [Mira y col., 1994]. La biología sin embargo no tiene ese problema.

## **5.2. Ruta de Datos**

Si pensamos en términos de neurofisiología inversa viendo a partir de un SN cuales podrían haber sido los objetivos de diseño y sus especificaciones funcionales nos encontramos con un conjunto de funciones que parecen análogas a lo que en electrónica digital y arquitectura de ordenadores conocemos como funciones de «ruta de datos», encargadas de facilitar el tráfico de datos desde y hasta los sitios específicos y en el tiempo adecuado para que se ejecute el «programa» de computación neuronal. Podemos distinguir, al menos, los siguientes procesos [Braitenberg, 1977; Bullock, 1979; Bullock y col., 1977; Schmitt y Worden, 1979]:

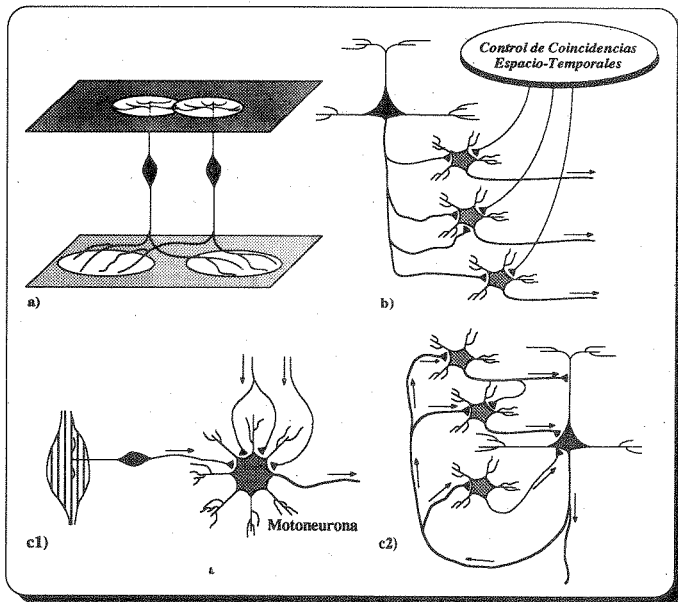
R.1. Convergencia y Divergencia masivas y con solape. Este aspecto de las arquitecturas neuronales proporciona el soporte anatómico para la integración de la información procedente de distintas fuentes y para la distribución selectiva de información local.

Las redes convergentes posibilitan la conversión de información espacial en información temporal (multiplexado), la detección de condiciones lógicas y la decisión local que puede usarse posteriormente como control excitatorio o inhibitorio de otras vías. Estas redes pueden considerarse entonces como mecanismos útiles en reconocimiento de caracteres (áreas sensoriales) y selección de acciones elementales en áreas motoras. En los procesos sensorio-perceptuales las redes convergentes pueden servir de soporte al código de primer orden donde la calidad de un estímulo (luz, color, sonido, ...) se representa mediante líneas etiquetadas.

Las redes divergentes poseen la estructura complementaria, son demultiplexoras y pasan la información serie a paralelo. Cumplen funciones de distribución de

información y son básicas para el cálculo paralelo. Ambas estructuras (convergentes y divergentes) no sólo aparecen en todas las regiones del sistema nervioso sino que de hecho ya existen sobre una única neurona.

La figura 14 ilustra estos tipos de red sobre algunos ejemplos sencillos. En 14.a se ilustra con dos neuronas vecinas la convergencia en sus campos receptivos, con solape en la información muestreada y la posterior divergencia en sus campos axónicos. En 14.b se muestra un proceso divergente a partir de varias colaterales del axón de una célula piramidal que usa interneuronas para facilitar el control adicional de la apertura o cierre de una u otra vía mediante la facilitación o inhibición de la interneurona correspondiente. La convergencia se ilustra en las figuras 14.c.1 y 14.c.2. En 14.c.1, el disparo de una motoneurona está controlado por la coincidencia espacio-temporal de un conjunto de señales que convergen sobre su campo dendrítico o sobre su soma. En 14.c.2 la convergencia se ilustra con una piramidal en la que distintas colaterales recurrentes del axón cierran lazos de realimentación a través de interneuronas que a su vez pueden recibir información de otras neuronas.



**Figura 14.** Convergencia-Divergencia masiva y solape de campos receptivos y axónicos. a) La misma información compartida por células vecinas. b) Divergencia. Varias colaterales de una misma neurona actúan en paralelo sobre distintas células diana (interneuronas). c1) Convergencia. La actividad de una neurona depende de la coincidencia temporal del estado de actividad de otras neuronas (aférentes específicas, interneuronas, fibras procedentes de otros grupos funcionales, etc.). c2) Convergencia sobre una piramidal de varios lazos de realimentación a través de interneuronas.

Otro aspecto importante de la convergencia-divergencia es la redundancia que conllevan debido al gran solape en los campos receptivos. Esto asegura tolerancia a fallos y aumenta la seguridad de funcionamiento. A lo largo de toda la revisión sobre las bases anatomo-fisiológicas de la computación neuronal se nos repite de forma insistente lo que podríamos llamar un «estilo genérico de computación» consistente en la detección, acumulación y olvido de miles de coincidencias espacio-temporales sobre circuitos redundantes. Una aplicación de este método de computación en visión [Fernández y Mira, 1992], junto con algunos ejemplos históricos [Kohonen, 1972] sugieren su posible valor en el campo de las RNAs.

R.2. Conducción pasiva (atenuada), activa (no atenuada) y modulada por tensión. Los procesos de conducción no se limitan al puro transporte pasivo e instantáneo de las señales. Existe conducción pasiva atenuada por la distancia y retardada de forma análoga a nuestras líneas de retardo analógico. Este hecho posee al menos dos consecuencias claras desde el punto de vista computacional. Por un lado los «pesos» de las conexiones deben atenuarse con la distancia [Caelli y col., 1993]. Por otro lado, el tiempo interviene como variable de cálculo de forma que durante la conducción también se realiza cálculo entre señales retardadas.

Existe también transmisión activa (no atenuada), con reconstrucción constante de la señal y modulación sobre impedancias no lineales cuyo valor depende del potencial en otro punto de la red.

Es evidente que en las RNAs la conducción tiene que ser pasiva. Por consiguiente, todas las otras propiedades observadas en la conducción biológica deben ser absorbidas por la computación local.

R.3. Conducción en sentido inverso (antidrómica). Junto al carácter unidireccional de las sinapsis en general, existe evidencia de conducción inversa, desde dendritas postsinápticas a axones presinápticos.

R.4. Control de la conducción por facilitación e inhibición absolutas. Además de los mecanismos de control de paso de información mediante interneuronas de forma análoga a la comentada en el apartado sobre convergencia-divergencia, existen mecanismos de facilitación y de inhibición absolutas, fuera del rango usual de sumar y restar en sinapsis excitadoras e inhibitoras, que están asociados a ciertos estados de actividad representados por terminales específicos y «privilegiados» o por la sintonía a neurotransmisores y otros procesos del nivel bioquímico que no encuentran una representación analítica en el nivel eléctrico. Por eso, para su descripción computacional necesitamos usar la algorítmica (condicional IF ... THEN ... de nuevo).

### 5.3. Computación Local

Es evidente que si existe computación en el sistema nervioso, en el sentido artificial del término, ésta debe estar distribuida por todo el tejido en un sentido holográfico. Sin embargo, si queremos hacer una vez más «neurofisiología inversa» y pensar en términos de arquitectura de ordenadores, podemos distinguir entre «ruta de datos» y «computación local», tanto a niveles subcelulares como al nivel neuronal y al de computación masiva. Centrándonos en el nivel neuronal, podemos encontrar al menos los siguientes tipos de procesos de computación local:

- CL.1. Transducción logarítmica.
- CL.2. Suma algebraica (excitación, inhibición).
- CL.3. Inhibición y facilitación pre y postsináptica multiplicativas. Multiplicadores analógicos.
- CL.4. Campos receptivos que definen estructuras centro-periferia y actúan como filtros activos espacio-temporales. Filtros sintonizados a neurotransmisores específicos.
- CL.5. Funciones de decisión no lineal con sensibilidad variable (pendiente) y valores absolutos adaptativos (aprendizaje). Disparo por derivada del estímulo.
- CL.6. Computación algorítmica específica, dependiente del nivel bioquímico.
- CL.7. Existencia de actividad aparentemente espontánea (situaciones de fatiga, falta de actividad, lazos locales de realimentación positiva, etc., disparo intermitente).
- CL.8. Máquinas híbridas con salidas múltiples (espigas y potenciales lentos) de una misma unidad que presenta multiplicidad funcional. Esto exige programación local que sólo puede llegar a través de las entradas.

### 5.4. Módulos Funcionales a Nivel Intermedio

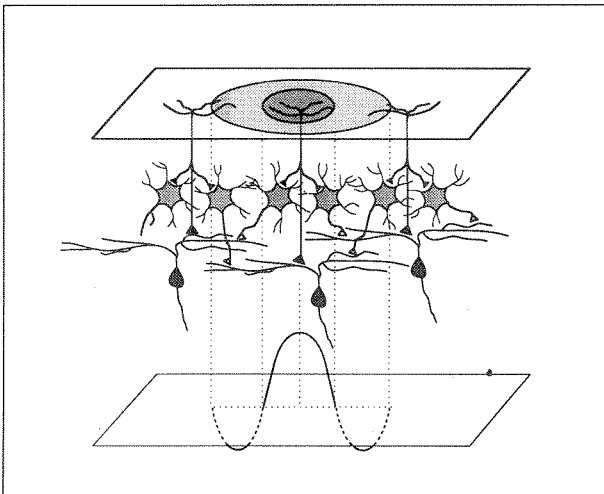
La computación local podría ser el equivalente a las puertas NAND o NOR en Electrónica Digital. Es decir, el conjunto completo de operadores básicos a partir de los cuales se puede sintetizar cualquier otra función, mediante el uso repetitivo de estos operadores y el esquema de conexión adecuado.

El siguiente nivel de integración en el SN parece ser un conjunto de circuitos locales que enlazan varias neuronas con una conectividad específica y que se repiten en prácticamente todos los SNs, tanto a nivel periférico como central. Vamos a comentar los siguientes:

- \* Inhibición lateral.
- \* Circuitos reverberantes.
- \* Arcos reflejos.
- \* Columnas.

MF1. Inhibición Lateral. Esta es probablemente la estructura que más se repite en el sistema nervioso y a su vez el módulo computacional sobre el que puede desarrollarse toda la computación local. Anatómicamente se corresponde con esquemas como el de la figura 15, en el que aparece una capa de neuronas del mismo tipo y con conectividad local tal que la respuesta de una unidad no sólo depende de sus entradas, sino de las entradas y respuestas de las células vecinas. En general, la interacción es de tipo inhibitorio de forma que la actividad de una neurona disminuye cuando están activas sus vecinas. Se distingue entre interacción recurrente y no recurrente y en ambos casos entre aditiva y multiplicativa-divisiva, de acuerdo con la operación matemática que mejor ajusta los resultados experimentales cuando se interpreta el código neuronal en términos de frecuencia de un tren de impulsos.

Desde el punto de vista computacional la función de las redes de interacción lateral es realizar una partición del espacio de entradas en tres regiones: centro, periferia y excluida, tal como comentamos al introducir el concepto de campo receptivo funcional (CRF). Lo mismo hace sobre las realimentaciones desde el espacio de salidas y en ambos casos realiza un proceso local sobre la zona central y otro sobre la zona periférica. Posteriormente compara el resultado de estos procesos y genera una decisión local que descarga en su espacio de salidas. Este proceso fue descrito por primera vez por Harline y Ratliff en el ojo compuesto del limulus [Hartline y Ratliff, 1957; Ratliff, 1965] y ha sido usado hasta la saciedad en el campo de las RNAs, tanto como filtro detector de contrastes espacio-temporales, como en redes autoorganizativas y en otros modelos algorítmicos tipo «el ganador se queda con todo».



**Figura 15.** Redes de interacción lateral. Estructura de la conexión no recurrente a través de interneuronas.

Lo interesante de estas redes es que existen prácticamente a todos los niveles en vertebrados e invertebrados, desde las primeras etapas de las vías sensoriales (retina) hasta las capas más centrales de la corteza cerebral. El otro aspecto interesante es que las redes de interacción lateral definen familias de computaciones que se proyectan en sus casos particulares al seleccionar el operador neuronal. De hecho, cambiando el término «inhibición» por el más general de «interacción», toda la computación neuronal dentro de una capa puede formularse mediante estos esquemas de interacción lateral. En el caso extremo de comportamiento lineal e invariancia ante traslaciones espaciales obtenemos sistemas de convolución espacio-temporal. Es decir, filtros espacio-temporales generalizados (pasa-baja, pasa-alta, pasa-banda y banda eliminada)

MF2. Circuitos Reverberantes. La realimentación múltiple positiva y negativa en todos y cada uno de los niveles (molecular, dendro-dendrito, neuronal, etc.) es otra de las estructuras computacionales que más se repiten en el SN. Constituyen la base del control, de varios mecanismos de memoria estática y dinámica (ciclos de oscilación) y de los comportamientos oscilatorios y de sincronización.

La realimentación puede darse (a) dentro de una misma neurona, mediante colaterales del axón que la reexcitan directamente o a través de interneuronas o (b), usando varias neuronas (cadenas, generadores de configuraciones, modos rítmicos espontáneos, etc.).

La inclusión de la realimentación tras el retardo en los modelos de neurona fue usual en la primera etapa [Caianiello, 1966; García Santesmases, 1976; McCulloch, 1965; Mira, 1971; Mira y Fonseca, 1970; Moreno-Díaz, 1971; Moreno-Díaz y McCulloch, 1968]. Posteriormente, en el renacimiento, se usaron principalmente redes sin realimentación. Afortunadamente, la recurrencia vuelve a ser incluida en las RNAs, aceptando un hecho evidente tanto en biología como en teoría de sistemas: sin retardo no hay computación secuencial ni oscilación.

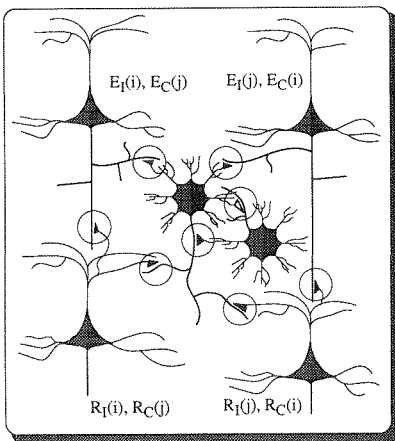
MF3. Arcos Reflejos. Otra de las estructuras básicas del sistema nervioso es el arco reflejo, que permite la asociación espacio-temporal de estímulos y es soporte de funciones de memoria asociativa y del aprendizaje por condicionamiento clásico. Su descripción formal es la siguiente.

En principio, existen ciertos estímulos del medio, correspondientes a la activación del axón de ciertas neuronas,  $\{E_i\}$ , que a su vez ponen en actividad un conjunto de interneuronas que provocan el disparo de otras neuronas (por ejemplo, motoneuronas) las cuales representan las respuestas innatas de la red,  $\{R_i\}$ . Un nuevo conjunto de estímulos,  $\{E_c\}$ , que en principio no estaban asociados a los procesos que provocan las respuestas innatas, si se presentan asociados espacial y temporalmente con  $E_i$ ,  $\{E_i, E_c\}$ , empiezan a provocar las mismas respuestas,  $\{R_i\}$ .

Es decir, el estímulo condicionado,  $E_c$ , se convierte en una señal de  $E_i$  que anuncia su probable presencia dando lugar a un proceso de categorización o abstracción que define clases de estímulos transitoriamente equivalentes. Cada respuesta,  $R_i$ , tiene asociada una clase de equivalencia de «tamaño» variable que incluye todos los estímulos que le son propios: los innatos y los condicionados. Tanto la creación como la extinción de las asociaciones son consideradas como respuestas adaptativas del organismo a los cambios del medio y se formulan a nivel de circuitos mediante una red de interneuronas que acumula las asociaciones y las olvida posteriormente, si no se refuerzan.

La estructura genérica de las redes que realizan la asociación espacio-temporal de estímulos, la acumulación de coincidencias persistentes y la posterior extinción, es la que se muestra en la figura 16. Cada neurona de entrada posee una vía directa a la neurona de salida y otra hacia interneuronas de asociación-acumulación. Cada neurona de respuesta se puede disparar por aferencias de la vía directa o por aferencias de la vía de asociación, siempre que el nivel de excitación acumulado supere un cierto valor umbral.

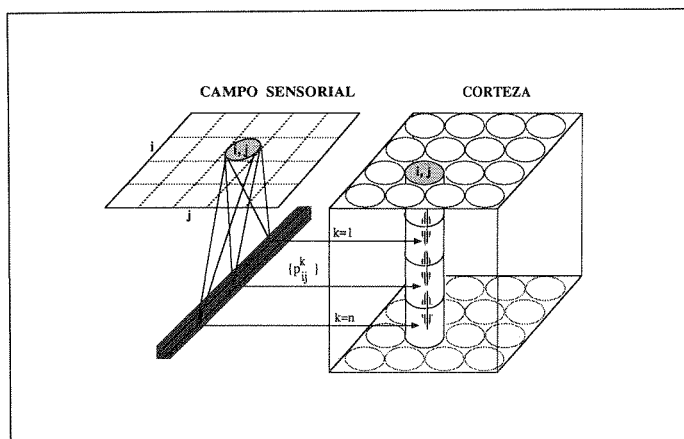
MF4. Columnas. Estudios en corteza primaria sensorial y motora ofrecen argumentos para una organización columnar de la corteza. Propuesta inicialmente por Lorente de No, son los trabajos de Mountcastle [1979], Werner y Whitsel [1973] y Hubel y Wiesel [Hubel y Wiesel, 1962, 1969, 1977], entre otros muchos, los que sugieren que las unidades funcionales básicas en corteza son las columnas, grupos verticales de 110 neuronas (aproximadamente). Esta organización permite la comparación simultánea de varias características de una misma zona del estímulo.



**Figura 16.** El reflejo condicionado como unidad fisiológica de la actividad nerviosa. Las interneuronas de asociación acumulan coincidencias espacio-temporales y abren y cierran vías de asociación.  $E_i(i)$ , es el estímulo de la vía directa (incondicionado) que produce  $R_i(i)$ .  $E_i(j)$  es el estímulo que se asocia para producir finalmente la misma  $R_i(i)$ . Dada la simetría de la red, el estímulo  $E_i(j)$ , innato para la respuesta  $R_i(j)$ , puede también ser sustituido tras la asociación por  $E_i(i)$ .

La figura 17 ilustra la correspondencia entre una matriz de datos de una modalidad sensorial (por ejemplo, visión o representación de la superficie del cuerpo) sobre la correspondiente matriz de corteza, manteniéndose la topología y las relaciones de vecindad con solape en el proceso de proyección cortical. La columna que hay debajo de un área de proyección (coordenadas  $i,j$ ) procesa información correspondiente a los distintos parámetros de posición, contraste, orientación, velocidad, dirección preferente y modalidad que definen el espacio de representación para la coordenada equivalente en el campo sensorial ( $i,j$ ). La columna integra estas coordenadas.

Además, entre columnas vecinas siguen existiendo los mismos mecanismos descritos anteriormente (MF1, MF2 y MF3) para circuitos locales con neuronas individuales. Es decir, hay inhibición lateral entre columnas vecinas. La actividad de una columna inhibe la actividad de las columnas que están en su periferia. Una estimación conservadora nos dice que tenemos del orden de 44.000 millones de estas columnas lo que nos da una idea de la capacidad computacional de la corteza sensorial.



**Figura 17.** Proyección sobre columnas en corteza que mantienen el orden topológico y procesan en paralelo distintos atributos del estímulo ( $p^k_{ij}$ ) correspondientes a cada coordenada matricial ( $i,j$ ) en el campo sensorial.

## 5.5. Funciones de Memoria

Está fuera del alcance de este trabajo un estudio razonablemente completo sobre las capacidades de memoria de la neurona, que van mucho más allá del mero almacén de datos y enlazan con los conceptos más generales de plasticidad y aprendizaje [Bullock, 1977; Hebb, 1949; Mira y Delgado, 1991b; Segev, 1992]. A título de resumen seleccionaremos los siguientes puntos:

- M1. Mecanismos moleculares y subcelulares (neuromoduladores, segundo mensajero, conductividad de membrana).

- M2. Mecanismos de habituación y cambios de sensibilidad.
- M3. Plasticidad sináptica tipo Hebb (por ejemplo, LTP en hipocampo), directa o a través de interneuronas detectoras de coincidencia.
- M4. Memoria analógica distribuida por retardo de transmisión sobre dendritas, soma y axón.
- M5. Retardo digital en la transmisión sináptica.
- M6. Mecanismos de adaptación en la evolución temporal del valor del umbral tras el disparo. Períodos refractarios absoluto y relativo.
- M7. Memoria estática tipo biestable.
- M8. Memoria dinámica tipo «registro de desplazamiento», donde la estructura del circuito reverberante almacena información a través de los distintos ciclos de oscilación en los que puede entrar bajo excitación constante.

## **5.6. Generadores de Patrones de Respuesta**

De nuevo la consideración en extenso de las capacidades computacionales de la neurona considerada como generadora de patrones de respuesta queda fuera del alcance de este trabajo [Bullock, 1979; Calvin y Graubard, 1979; Steriade y Llinás, 1988]. Sí que queremos señalar sin embargo que junto a las formas de respuesta usadas en RNAs (potenciales lentos y trenes de espigas) debe considerarse a la neurona como un generador algorítmico de patrones de respuesta con salida múltiple y comportamiento modal que incluye el modo «tren de espigas», el modo «oscilador» con ritmos variables y el modo «espigas extra», entre otros, cambiando de uno a otro bajo la influencia de los distintos niveles, en particular del molecular. Existen, al menos, los siguientes modos:

- G1. Modo salida única y potencial lento.
- G2. Modo salida única y tren de espigas con modulación lineal (convertidores tensión-frecuencia).
- G3. Modo salida múltiple (G1 y G2).
- G4. Modo marcapasos. Generación autónoma de ritmos con modulación en frecuencia y mecanismos de «reset» e inicio.
- G5. Modo coordinado. Sincronización masiva y patrones coordinados de control.

De acuerdo con el plan general de este trabajo hemos visto ya las analogías y diferencias entre la computación neuronal y la simbólica y las exigencias computacionales mínimas de los modelos de neurona y red neuronal que pretenden parecerse a lo que conocemos de las neuronas reales. Vistas las bases anatómo-fisiológicas a nivel de arquitecturas, ruta de datos, computación local, memoria y producción de salidas, la verdad es que no nos parecen tan «mínimas» esas exigencias.

Cuando pasamos de la biología al mundo de lo artificial es interesante recordar una vez más las limitaciones del silicio frente a las del tejido neuronal. Por eso es evidente que no podremos transportar todas las propiedades enumeradas anteriormente (A1, A2, ..., G1, ..., G5) al campo de las RNAs. Quedan sin embargo puertas abiertas para mejorar nuestros modelos de computación local, las arquitecturas de las redes y los mecanismos de aprendizaje. En particular, la autonomía, la expansión no lineal de los espacios de entrada y realimentación, la recurrencia con estructuras locales tipo inhibición lateral, la autoorganización, los mapas de corteza, la tolerancia a fallos y las arquitecturas de grupos funcionales especializados.

## 6. ASPECTOS METODOLÓGICOS

De acuerdo con el plan propuesto para este trabajo, queremos comentar ahora el tercer punto que nos preocupa dentro de la computación neuronal. Nos referimos a la metodología o, más exactamente, a la falta de metodología en el análisis y diseño de RNAs. El apartado anterior sobre fundamentos biológicos puede considerarse también como fuente de inspiración metodológica. Vamos a desarrollar este apartado de forma resumida de acuerdo con los siguientes puntos:

- 6.1. Complejidad y autonomía del modelo de neurona.
- 6.2. Arquitecturas multicapa de clasificadores genéricos con diversidad neuronal.
- 6.3. Metodología previa a la elección de una solución neuronal.
- 6.4. Distinción entre «red ciega» y «red basada en conocimiento» y defensa de esta última alternativa.

### 6.1. Complejidad y autonomía del modelo de neurona

Si queremos transportar al campo de las RNAs algunas de las especificaciones funcionales que hemos visto en el apartado anterior, la primera exigencia es aceptar la autonomía de los procesadores y una mayor complejidad en el modelo de neurona usado. Debe ser un circuito autoprogramable para una familia de computaciones locales y algoritmos de aprendizaje. Su estructura interna debe permitir el flujo de "datos de cálculo" y "datos de autoprogramación". El control de distintos modos de funcionamiento (inicialización, uso, entrenamiento) y la tolerancia a fallos deben caber también en la arquitectura del módulo y en sus facilidades de comunicación. La figura 18 ilustra el aspecto cualitativo de la arquitectura de este módulo autónomo. Parece obvio que la complejidad inherente a esta propuesta es muy superior a la del sumador seguido de sigmoide. Estamos hablando de complejidades análogas a las

de un microprocesador. Los detalles de cada implementación dependerán del modelo de computación local y de los algoritmos de aprendizaje seleccionados. Finalmente, la computación local y el aprendizaje dependerán de la situación contextual en la red y de la aplicación (visión, control, predicción, comunicación, etc...).

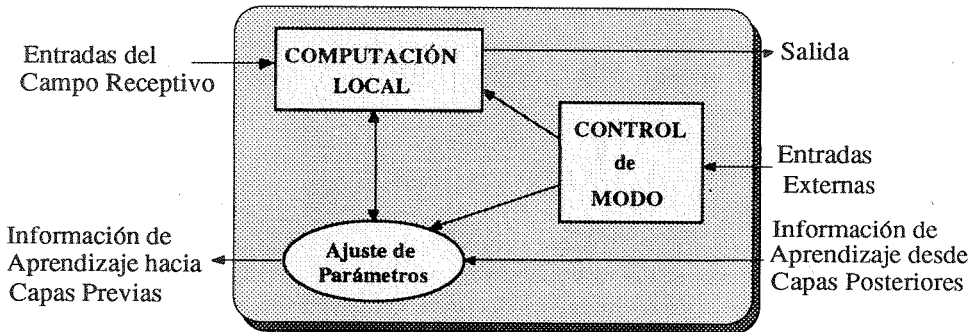


Figura 18. Módulo autónomo de computación neuronal.

Tal como hemos comentado a lo largo de toda la segunda parte del trabajo, es esencial recordar que deben existir muchos tipos de neuronas. Por consiguiente, es difícil prever en el estado actual del conocimiento, si una única estructura podría incluirlas a todas o no. Estamos hablando de familias de circuitos de aplicaciones específicas (ASICs) programables que pudieran en un momento dado del desarrollo dar paso a «microprocesadores autoprogramables». La tecnología actual lo permite. El problema está en la falta de propuestas a nivel de diseño.

Afortunadamente, un primer paso ya se está dando. Hay una cierta tendencia en el campo de las implementaciones digitales [Sánchez-Sinencio y Newcomb, 1993; Treleaven y col., 1989; Vellasco y Treleaven, 1993; Vidal, 1988; Wawrzynek y col., 1993] a incluir todos los procesos dentro del módulo de computación local, permitiendo su desconexión del ordenador. Ya hemos comentado anteriormente que las dos formas de extender la computación neuronal siguiendo las sugerencias de la biología eran analíticas o inferenciales. Manteniéndonos dentro del campo de las «neuronas» analíticas, las extensiones que parecen evidentes y posibles son:

- 6.1.1. La expansión no lineal de los espacios de entrada (interacción de aferentes) y salida (contactos axón-axón), incluyendo términos polinómicos, logarítmicos y señales moduladas.
- 6.1.2. El muestreo del campo de datos sobre memorias FIFO, de acuerdo con la forma espacio-temporal del campo receptivo funcional (CRF), organizado en

- volúmenes concéntricos alternativamente excitadores e inhibidores que incluyen el tiempo como otra variable de cálculo.
- 6.1.3. La realimentación de respuestas retardadas dentro de cada capa (grupo funcional) y entre capas diferentes. Esto supone introducir una nueva memoria local, también tipo FIFO, y definir campos dendríticos basales. La conectividad recurrente con neuronas vecinas queda incluida en la arquitectura a través del volumen de muestreo en este espacio de salidas.
  - 6.1.4. La ampliación del campo de operadores locales que junto a la usual suma de productos, puede incluir la lógica de extremos y otros operadores sencillos que permitan la autoprogramación local.
  - 6.1.5. Las facilidades estructurales necesarias para el control de modo de funcionamiento y para el aprendizaje.
  - 6.1.6. Si quitamos las exigencias de los modelos analíticos, las neuronas artificiales se convierten en condicionales hardware, tal como comentamos en la primera parte del trabajo. La arquitectura de condicionales hardware permite incluir los apartados anteriores (6.1.1, 6.1.2, 6.1.3 y 6.1.4) y el concepto más amplio de generador algorítmico de patrones de respuesta, tal como comentamos en el apartado 5.6.
  - 6.1.7. La conectividad sobreabundante y la poda selectiva tras el entrenamiento.

## **6.2. Arquitecturas multicapa de clasificadores genéricos con diversidad neuronal**

La siguiente sugerencia que la biología hace al campo de las RNAs es la idea de red genérica, con funciones de clasificación, organizada en distintos grupos funcionales (capas) que calculan de forma concurrente y/o secuencial y que incluye distintos tipos de neurona (tantos como capas, al menos).

Si pensamos en las redes como clasificadores genéricos, las distintas capas están asociadas a la descomposición del problema en distintas tareas tales como extracción de características, medidas de distancias a clases de equivalencia o cálculo de máximos o mínimos, para seleccionar la clase más próxima. Para cada una de estas tareas debe proporcionarse una arquitectura local adecuada hasta llegar a la subtarea no reductible, que es la neurona. Vimos que en biología existían neuronas transductoras, extractoras de propiedades, efectoras, de asociación y de integración. En el campo de las RNAs, esto es sólo una pista y es claro que los distintos tipos de neuronas necesarias para la solución de un problema de clasificación dependerá fuertemente de la descomposición que hagamos del procedimiento de solución.

### **6.3. Metodología previa a la elección de una solución neuronal**

Parece evidente, aunque se olvida frecuentemente, que las RNAs no son la mejor solución para todos los problemas computacionales. Por consiguiente ante un problema concreto es necesario realizar primero una valoración de las exigencias computacionales del mismo y un estudio de viabilidad de la solución neuronal, ya sea para ser usada de forma exclusiva o en cooperación con otras técnicas de computación numérica o simbólica. La computación simbólica puede actuar como preprocesador o como coprocesador de la neuronal y viceversa.

Como norma general, una solución analítica es preferible a una algorítmica, la algorítmica es preferible a la heurística y ésta a la neuronal. También es norma general que el carácter variable e impredecible de un proceso junto con la necesidad de adaptación en tiempo «real», aconseja la solución neuronal. Veamos de forma algo más precisa un conjunto de cuestiones que nos pueden guiar en la elección de la solución neuronal para un determinado problema:

C.1 ¿Cuáles son las exigencias computacionales del problema?. Es decir, qué tipo de tarea hay que ejecutar (clasificación, decisión, control, asociación espacio-temporal, etc.) y cómo se puede hacer computacional, independientemente del tamaño del «grano» y de la fuente de programación (externa o autónoma).

Veremos a continuación la distinción entre «red ciega» y «red basada en conocimiento». Todo el conocimiento del problema, en el que se va a basar su diseño inicial, debe elicitar dentro de este apartado (C.1). La red neuronal es un sistema de aprendizaje que debe describirse en dos dominios. El dominio propio del observador (DO) y el dominio propio de los procesadores neuronales (DP). En el DO se encuentra la descripción del problema usando lenguaje natural y el metalenguaje de la tarea específica. Aquí hay que definir las funcionalidades que se esperan de la red neuronal. Hay que especificar también el espacio de representación (las señales de entrada y sus significados), y el conocimiento inyectado que no está en las neuronas pero que es imprescindible para comprender el significado de su computación. El siguiente paso es ver si el DP adecuado es neuronal, simbólico o mixto.

C.2 ¿Es posible y aconsejable usar redes neuronales para su solución?. Es decir, ¿es segmentable el problema y por consiguiente admite una solución distribuida y en este caso, es conveniente usarla?. Dicho de otro modo, ¿la naturaleza del problema posee las características de modularidad, computación masiva, tiempo real, medio cambiante, exigencia de ajuste de parámetros y autonomía de procesadores, que hacen aconsejable una solución neuronal?.

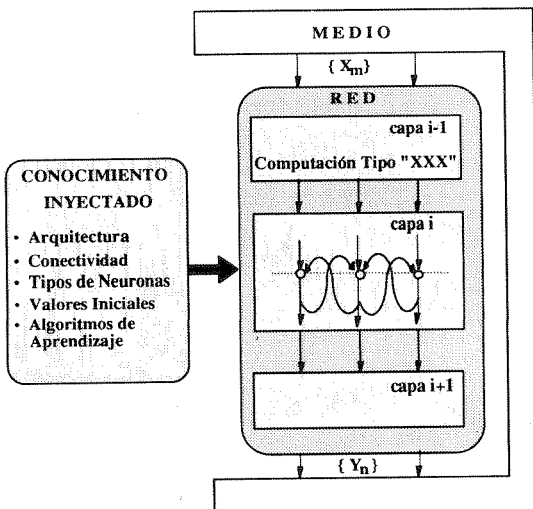
Visto el problema desde el campo general del aprendizaje (simbólico o conexionista), la cuestión a aclarar, en términos de un conjunto de criterios es la

conveniencia de una formulación inductiva (tipo ID3), deductiva (tipo EBL), conexionista (tipo retropropagación del gradiente o asociativa) o mixta.

C.3 Supuesta afirmativa la contestación a las cuestiones C.1 y C.2, ¿cuál es la arquitectura inicial de la red? Una arquitectura muy general es la que se ilustra en la figura 19, procede de las propuestas iniciales de W.S. McCulloch [1965], R. Moreno Díaz [Moreno-Díaz y Rubio, 1979], y A.E. Delgado [Delgado, 1978; Delgado y Mira, 1979; Delgado y col., 1989] y está organizada en capas especializadas con computación recurrente y paralela dentro de cada capa y secuencial entre capas sucesivas.

Esta formulación es muy general, pensando en su posterior implementación y no incluye las tendencias actuales en arquitecturas incrementales. Por eso esta propuesta debe considerarse sólo como una de las opciones posibles. Es decir, pueden existir problemas que aconsejen otras arquitecturas e incluso, tal como comentamos anteriormente soluciones híbridas, con parte de computación conexionista y parte algorítmica, en el sentido convencional.

C.4 Contestada C.1 y C.2 y propuesta una arquitectura en C.3, ¿Cuál es el tipo de computación local especializada correspondiente a cada una de las capas?.



**Figura 19.** Arquitectura multicapa que intenta poner de manifiesto la distinción entre la computación de la red y el conocimiento inyectado desde el exterior al seleccionar el tipo de neurona para cada capa, la conectividad, los valores iniciales en los «pesos» (en el modelo del sumador) o en los valores de los parámetros de la computación local (para otros modelos distintos, en particular, para los basados en conocimiento) y la síntesis local de los algoritmos de aprendizaje.

Este punto es esencial en la nueva perspectiva que hemos querido darle en este trabajo a la computación neuronal. Está relacionado con el concepto biológico de especialización funcional y con la idea de «red basada en conocimiento», como alternativa a las redes ciegas.

La clave está en decidir a qué computación le llamamos «neurona artificial». Desde la propuesta más usual (sumador más sigmoide) hasta el otro extremo (microprocesador autoprogramable), hay todo un repertorio de soluciones posibles.

C.5 No hay computación neuronal sin aprendizaje, por consiguiente la estructura de cálculo propuesta en C.4 debe autoprogramarse. Al ser autónomos los módulos, la autoprogramación debe ser local. Al existir capas especializadas, el algoritmo debe depender de la función de la capa (por ejemplo, extracción de características, métricas, selección de máximos, etc.).

El tema del aprendizaje y su síntesis local no ha sido tratado en este trabajo con la profundidad que se merece. Sólo hemos repetido la idea cualitativa del límite en la complejidad de la función local si ésta debe ser autoprogramable.

#### **6.4. Distinción entre «red ciega» y «red basada en conocimiento» y defensa de esta última alternativa**

En la revisión de las bases neurofisiológicas hemos visto que nada parece arbitrario en el diseño que la evolución ha hecho de nuestros sistemas nerviosos. También es un hecho obvio que en computación simbólica o numérica, y en particular en el campo de los SSBCs, todo el conocimiento que se tiene inicialmente sobre un problema se usa para programar su solución.

Sorprendentemente, en el campo de la computación neuronal se usan en general «redes ciegas», con alto grado de conectividad y con la misma función para todas las neuronas. Además, se suele aconsejar iniciar el proceso de aprendizaje usando valores pequeños y aleatorios para todos los pesos para mejorar las condiciones de convergencia en el entrenamiento.

La sugerencia más evidente desde la biología es justo la opuesta. Es decir, debemos usar todo el conocimiento inicial sobre el problema y los posibles métodos de solución para definir la arquitectura de la red. Por poner un ejemplo elemental, si tenemos un problema de identificación o control de un sistema lineal de orden  $n$ , en vez de incrementar el número de capas ocultas o el número de

neuronas por capa para conseguir un mejor ajuste de la respuesta del sistema, parece más razonable usar una red cuya arquitectura refleje la estructura del problema y cuyas neuronas dejen al aprendizaje sólo los parámetros cuyo valor no se conoce inicialmente.

Para todos los sistemas lineales, podemos admitir que existe una representación de su dinámica en términos de un conjunto de ecuaciones en diferencias finitas:

$$A(q^{-1}) y(t) = B(q^{-1}) u(t-1)$$

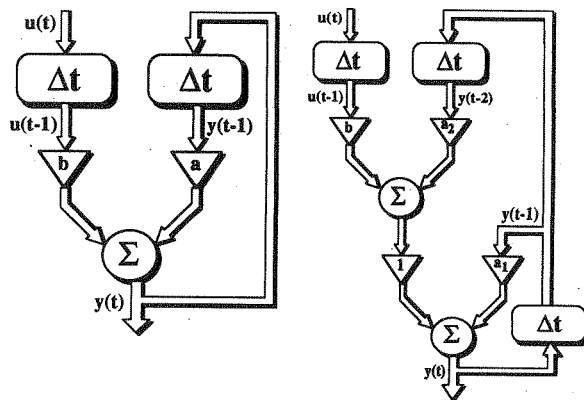
$$A(q^{-1}) = 1 + a_1 q^{-1} + \dots + a_n q^{-n}$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + \dots + b_m q^{-m}$$

donde A y B son polinomios y  $q^{-1}$  es el operador de diferencias. Parece natural abordar estos problemas usando neuronas de primer orden o de segundo orden, como las ilustradas en la figura 20, en vez de la suma convencional.

Para las de primer orden, los parámetros a y b cumplen la función de los pesos sinápticos y la sigmoide de salida se sustituye por la ecuación de una recta,  $y=x$ , con una primera derivada constante e igual a la unidad. Para las neuronas de segundo orden (figura 20.b), podemos usar dos de primer orden de forma que ahora los parámetros son (b,  $a_2$  y  $a_1$ ).

Obsérvese que en este caso los parámetros (pesos, funciones de decisión y operación local) representan parámetros estructurales del sistema real, con significado físico. Cuando se usan estos parámetros estructurales el entrenamiento es mucho más eficiente [Mira y col., 1994] y el aprendizaje termina por identificar al sistema.



**Figura 20.** Ejemplo de neuronas basadas en el conocimiento inicial sobre la estructura de los sistemas lineales. a) Neurona de primer orden que calcula  $y(t) = a y(t-1) + b u(t-1)$ . Sus parámetros de aprendizaje son a y b. b) Neurona de segundo orden. Sus parámetros son (b,  $a_1$  y  $a_2$ ).

Hemos usado este ejemplo sencillo para ilustrar las ventajas asociadas a las redes basadas en conocimiento. Es evidente sin embargo el amplio espectro de posibilidades que abre esta metodología ante problemas reales de mayor complejidad.

## **CONCLUSIONES**

Hemos presentado en este trabajo los tres puntos que más nos preocupan en computación neuronal. a) Su relación con la computación simbólica. b) La reflexión sobre las bases anatomo-fisiológicas y la evidencia que de allí emana acerca de la complejidad necesaria en los modelos neuronales y c) la falta de metodología en el campo y la sugerencia de algunos consejos para el desarrollo de una aplicación concreta. En particular, el uso del conocimiento previo para definir la arquitectura inicial de la red.

Es nuestro sentimiento profundo que la actitud ante estos tres puntos va a ser determinante del futuro de la computación neuronal.

El establecimiento del puente simbólico-conexionista va a permitir integrar la Inteligencia Artificial con las RNAs, tomando lo mejor de ambos paradigmas. No hay competición sino complementariedad entre técnicas y estrategias distintas.

El volver de nuevo los ojos hacia la Biología (como en los orígenes del campo) va a ser fructífero en varios sentidos. Por un lado aumentará nuestra comprensión de lo vivo. Por otro encontraremos una fuente inagotable de inspiración sobre nuevos estilos de computación y, finalmente, los resultados obtenidos nos permitirán el desarrollo de soluciones más eficientes a una amplia variedad de problemas técnicos (sobre todo si no olvidamos la complementariedad propuesta en el primer punto).

Finalmente, el intentar introducir metodología y desarrollar redes basadas en conocimiento nos va a permitir dotar a la computación neuronal de un cuerpo de conocimientos y técnicas intermedias análogo al existente actualmente en la informática convencional.

Inversamente, y muy a nuestro pesar, si no se producen estos cambios de actitud el modelo de computación por sumadores y sigmoides es probable que pronto quede «exhausto» y el campo empiece a invernar de nuevo durante otros 20 años, como pasó en los anteriores renacimientos, desde 1943. Seamos optimistas y empujemos en la dirección que creemos adecuada para evitar esta última alternativa.

## **AGRADECIMIENTOS**

Agradecemos el soporte económico de la CICYT a través de los proyectos TIC 92/136 y TIC 94/0095 en cuyo contexto se han desarrollado los trabajos que aquí se presentan.

## REFERENCIAS

- Anderson, J.A. y Rosenfeld, (Eds.), *Neurocomputing: Foundations of Research*, The MIT Press, Cambridge, 1989.
- Anderson, J.A., Pellionisz, A. y Rosenfeld, (Eds.), *Neurocomputing 2: Directions for Research*, The MIT Press, Cambridge, 1989.
- Beach, F.A. y col., (Ed.), *The neuropsychology of Lashley*, McGraw-Hill, 1960.
- Blum, M., «Properties of a Neuron with Many Inputs», en Foester, H. von y Gopf, G. (Eds.), *Principles of Self Organization*, MacMillan, Pergamon, N. York, 1961.
- Boticario, J.G. y Mira, J., «Some Methodological Aspects of Machine Learning», *Cybernetic & Systems: An International Journal*, Vol. 25, N. 2, marzo-abril, (1994), 233-258.
- Braitenberg, V., *On the Texture of Brains: An Introduction to Neuroanatomy for the Cybernetically Minded*, Springer-Verlag, New York, 1977.
- Bullock, T.H., «Evolving Concepts of Local Integrative Operations in Neurons», en Schmitt, F.O. y Worden, F.G. (Eds.), *The Neuroscience Fourth Study Program*, The MIT Press, Cambridge, Mass, 1979, 43-49.
- Bullock, T.H. y col., *Introduction to Nervous Systems*, W.H. Freeman and Company, San Francisco, 1977.
- Caelli, T.M., Squire, D.M. y Wild, T.P.J., «Model-based Neural Networks», *Neural Networks*, Vol. 6, N. 5, (1993), 613-625.
- Caianiello, E., «Decision Equations and Reverberations», *Kybernetik*, 3, (1966), 98-100.
- Calvin, W.H. y Graubard, K., «Styles of Neural Computation», en Schmitt, F.O. y Worden, F.G. (Eds.), *The Neuroscience Fourth Study Program*, The MIT Press, Cambridge, Mass, 1979, 513-524.
- Carpenter, G.A., «A Geometric Approach to Singular Perturbation Problems with Applications to Nerve Impulse Equations», *Journal of Differential Equations*, Vol. 23, (1977), 335-367.
- Carpenter, G.A. y Grossberg, S. (Eds.), *Neural Networks for Vision and Image Processing*, A Bradford Book, Camb., M.A., 1992.
- Clarkson, T.G. y Ng, C.K., «Architecture for Self-learning Neural Network Modules», en Mira, J., Cabestany, J. y Prieto, A. (Eds.), *New Trends in Neural Computation*, LNCS 686, Springer-Verlag, 1993, 471-475.

- De Felipe, J. y Fariñas, I., «The Pyramidal Neuron of the Cerebral Cortex: Morphological and Chemical Characteristics of Synaptic Inputs», *Progress in Neurobiology*, Vol. 39, (1992), 563-607.
- Delgado, A.E., *Modelos Neurocibernéticos de Dinámica Cerebral*, Tesis Doctoral, ETSIT, Madrid, 1978.
- Delgado, A.E. y Mira, J., «Residual Function After Lesion in Formal Neural Nets», en Lainiotis, D.G. y Tzannes, N.S.D. (Eds.), *Applications of Information and Control Systems*, Vol. 3, Reidel Publishing Company, Dordrecht, Holland, 1979, 209-216.
- Delgado, A.E., Mira, J. y Moreno-Díaz, R., «A Neurocibernetic Model of Modal Co-Operative Decision in the Kilmer-McCulloch Space», *Kybernetes*, Vol. 18, N. 3, (1989), 48-57.
- Dietterich, T.G. y Flann, N.S., «A Study of Explanation-Based Methods for Inductive Learning», *Machine Learning*, 4, (1989), 187-226.
- Fahlman, S.E. y Hinton, G.E., «Connectionistic Architectures for Artificial Intelligence», *Computer*, Vol. 20, N. 1, (1978), 100-109.
- Fernandez, M.A. y Mira, J., «Permanence Memory: A System for Real Time Motion Analysis in Image Sequences», *Proceedings of the LAPR Workshop on Machine Vision Applications*, Tokio, (1992), 249-252.
- Fonseca, J.S. da, Mira, J. y Barata, J., «Learning as a Change of Intention and of Signification», en Fonseca, J.S. da (Ed.), *Signification and Intention*, Universidad de Lisboa, 1970, 31-49.
- Fukushima, K., «Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position», *Biological Cybernetics*, Vol. 36, (1980), 139-202.
- Gallant, S.I., *Neural Network Learning and Expert Systems*, A Bradford Book, The MIT Press, Camb., MA., 1993.
- García Santesmases, J. y Moreno Díaz, R., «Algunas Analogías Electrónicas de Elementos Neuronales», *Anales de la Real Sociedad de Física y Química*, Serie A, Tomo LIX(a), (1963), 251-260.
- García Santesmases, J., Rubio, F. y Mira, J., «Modelo Electrónico de Inhibición Lateral en el Ojo de Límulus», *Anales de Física. Real Soc. Esp. de Física y Química*, Tomo LXV, (1969), 219-234.

- García Santesmases, J., Mira, J., Rubio, F. y González, T., «Hacia una Teoría del Aprendizaje en Redes Neuronales», *Revista de Automática*, Madrid, N. 10, octubre-noviembre, (1970a), 11-27.
- García Santesmases, J. y col., «Harmonic Analysis in the Visual Pathway of the Higher Mammals», en Rose (Ed.), *Progress of Cybernetics*, Vol. II, Gordon and Breach, London, 1970b, 403-418.
- García Santesmases, J., Moreno-Díaz, R., Rubio, F. y Mira, J., «Formal Neural Nets», *Int. Conf. on Information Sciences and Systems*, Univ. of Patras, Patras, Greece, agosto, 1976, 19-24.
- Gonzalo, J., «Las Funciones Cerebrales Humanas Según Nuevos Datos y Bases Fisiológicas», *Trabajos del Instituto Cajal de Inv. Biol.*, Vol. XLIV, Madrid, (1952).
- Grossberg, S., «A Neural Theory of Punishment and Avoidance, Part I: Qualitative Theory», *Mathematical Biosciences*, N. 15, (1972), 39-67.
- Hartline, H.K. y Ratliff, F., «Inhibitory Interaction of Receptor Units in the Eye of Limulus», *J. Genl. Physiol.*, 40, (1957), 357-376.
- Hebb, D.O., *The Organization of Behavior*, J. Wiley, N. York, 1949.
- Hecht-Nielsen, R., «Neurocomputing: Picking the Human Brain», *IEEE Spectrum*, marzo, (1988), 36-41.
- Hopfield, J.J., «Neural Networks and Physical Systems with Emergent Collective Computational Abilities», *Proceedings of the National Academy of Sciences*, Vol. 79, (1982), 2554-2558.
- Hopfield, J.J. y Tank, D.W., «Computing with Neural Circuits: a Model», *Science*, Vol. 233, (1986), 625-633.
- Hubel, D.H. y Wiesel, T.N., «Receptive Fields, Binocular Interaction and Functional Architecture in the Cats Visual Cortex», *J. Physiol.*, Londres, 166, (1962), 106-154.
- Hubel, D.H. y Wiesel, T.N., «Anatomical Demonstration of Columns in the Monkey Striate Cortex», *Nature*, 221, (1969), 747-750.
- Hubel, D.H. y Wiesel, T.N., «Functional Architecture of Macaque Monkey Cortex», *Proc. R. Soc. Lond*, B198, (1977), 1-59.
- Hwang, K. y col., «Computer Architectures for A.I. Processing», *Computer*, Vol. 20, N. 1, (1987), 19-27.

- Kilmer, W.L., McCulloch, W.S. y Blum, J., «Some Mechanisms for a Theory of the Reticular Formation», en Mesarovic, M. (Ed.), *System Theory and Biology*, Springer-Verlag, New York, 1968.
- Kleene, S.C., «Representation of Events in Nerve Nets and Finite Automata», en Shannon, C.E. y McCarty, J. (Eds.), *Automata Studies*, Princeton Univ. Press, Princeton, 1956, 3-42.
- Klimasauskas, C.C., *The 1989 Neuro-Computing Bibliography*, A Bradford Book, The MIT Press, Cambridge, Mass, 1989.
- Koch, C. y Poggio, T., «Biophysics of Computation: Neurons, Synapsis and Membranes», en Edelman, Gall y Cowan, (Eds.), *New Insights Into Synaptic Function*, J. Wiley and Son, 1985.
- Kohonen, T., «Correlation Matrix Memories», *IEEE Transactions on Computers*, Vol. C-21, (1972), 353-359.
- Kohonen, T., «The 'Neural' Phonetic Typewriter», *Computer*, Vol. 2, N. 3, (1988), 11-22.
- Kohonen, T. y Fogelman-Soulie, F. (Eds.), *Cognitiva-90*, North-Holland, 1991.
- Kung, S.Y. y Hwang, «A unified Systolic Architecture for Artificial Neural Networks», *Journal of Parallel and Distributed Computing*, 6, (1989), 358-387.
- Lee, T-C., *Structure Level Adaptation for Artificial Neural Networks*, Kluwer Acad. Pub., Boston, 1991.
- Lerma, M.A., *Nuevos Procedimientos Heurísticos-Constructivos para el ajuste de Redes de Neuronas*, Tesis Doctoral, Fac. de Informática, Univ. Politécnica de Madrid, 1991.
- Lettvin, J.Y. y col., «What the Frog's Eye Tells the Frog's Brain», *Proceedings of the IRE*, Vol. 47, N. 11, (1959), 1940-1959.
- Lippmann, R.P. y col., *Advances in Neural Information Processing Systems 3*, Morgan Kaufmann Publishers, 1991.
- Luria, A.R., *El cerebro en Acción*, Ed. Fontanella, Barcelona, 1974.
- Marr, D. y Poggio, T., «Cooperative Computation to Understanding Neural Circuitry», *Science*, 194, (1976), 283-287.
- McCulloch, W.S. y Pitts, W., «A Logical Calculus of the Ideas Immanent in Nervous Activity», *Bulletin of Mathematical Biophysics*, Chicago Univ. Press, Vol. 5, (1943), 115-133.

- McCulloch, W.S., «The Reliability of Biological Systems», en Yovits, M.C. y Cameron, S. (Eds.), *Self-Organizing Systems*, Pergamon Press, Oxford, 1960, 264-281.
- McCulloch, W.S., *Embodiments of Mind*, The MIT Press, Mass., 1965.
- Michalski, R.S., Carbonell, J.G. y Mitchell, T.M., (Eds.), *Machine Learning. An Artificial Intelligence Approach*, Springer-Verlag, Berlín, 1984.
- Mitchell, T.M., Keller, R.M. y Kedar-Cabelli, S.T., «Explanation-Based Learning: A Unified View», *Machine Learning*, Vol. 1, (1986), 47-80.
- Minsky, M.L. y Papert, S., «Linearly Unrecognizable Patterns», en *Mathematical Aspects Of Computer Science*, Proc. Symp. Appl. Math. (1967), 19, 47,217.
- Minsky, M.L. y Papert, S., *Perceptrons. An Introduction to Computational Geometry*, The MIT Press, Camb., Mass., 1969.
- Mira, J., *Modelos Cibernéticos de Aprendizaje*, Tesis Doctoral, Fac. de Ciencias, Universidad de Madrid, 1971.
- Mira, J., «System Behaviour and Computing Structure», Pichler, F. y Moreno Díaz, R. (Eds.), *Computer Aided Systems Theory*, LNCS 585, Springer-Verlag, Berlín, 1989, 267-283.
- Mira, J. y Delgado, A.E., «A Logical Model of Co-operative Processes in Cerebral Dynamics», *Cybernetics and Systems: An International Journal*, 18, (1987), 319-349.
- Mira, J. y Delgado, A.E., «On Processors and Processes in Neural Computing Machines», *Proc. Int. Cong. on Knowledge Engineering*, Madrid, (1988), 201-214.
- Mira, J. y Delgado, A.E., «Always Trying to Write an Equation for the Brain», Prieto, A. (Ed.), *Artificial Neural Networks*, LNCS 540, Springer-Verlag, Berlín, 1991a, 93-100.
- Mira, J. y Delgado, A.E., «Linear and Algorithmic Formulation of Co-operative Computation in Neural Nets», Pichler, F. y Moreno Díaz, R. (Eds.), *Computer Aided Systems Theory*, LNCS 585, Springer-Verlag, Berlín, 1991b, 2-20.
- Mira, J. y Fonseca, J.S. da, «Neural Nets from the Viewpoint of Signification and Intention», en Fonseca, J.S. da (Ed.), *Signification and Intention*, Faculdade de Medicina, Universidad de Lisboa, 1970.
- Mira, J., Lopez-Aligue, F. y Suarez, T.G., «Aprendizaje en Redes Probabilísticas», *Actas del Congreso Nacional de Automática*, Barcelona, Tomo II, 1972, 745-773.

- Mira, J., Delgado, A.E. y Moreno-Díaz, R., «Cooperative Processes in Cerebral Dynamic», en Lainiotis, D.G. y Tzannes, N.S. (Eds.), *Applications of Information and Control Systems*, Reidel Pub. Co. Dordrecht, 1980, 273-280.
- Mira, J., Delgado, A.E., Zapata, E.L. y Cabello, D., «On The Lesion Tolerance Problem for Co-Operative Processes», Richter, L., Le Beux, P., Chroust, G. y Noguez, G. (Eds.), *Implementing Functions: Microprocessors and Firmware*, North-Holland Publishing Company, Amsterdam, 1981, 71-80.
- Mira, J. y col., «A Theoretical Proposal to Embody Co-operative Decision in the Nervous System», *International Conference on World Problems and Systems Learning*, Detroit, Michigan, 1983, 23-27.
- Mira, J. y col., *Computación Neuronal*, Curso de tercer Ciclo de la UNED, Madrid, 1992.
- Mira, J., Cabestany, J. y Prieto, A. (Eds.), *New Trends in Neural Computation*, LNCS 686, Springer-Verlag, Berlin, 1993a.
- Mira, J., Delgado, A.E., Marín, R. y Barro, S., «Computación Neuronal Borrosa», en Sobrino, A. y Barro, S. (Eds.), *Estudios de Lógica Borrosa y sus Aplicaciones*, Cap. 8, Pub. de la Universidad de Santiago, 1993b, 197-224.
- Mira, J. y col., «Towards More Realistic Self Contained Models of Neurons: High-order Recurrence and Local Learning», en Mira, J., Cabestany, J. y Prieto, A. (Eds.), *New Trends in Neural Computation*, LNCS 686, Springer-Verlag, 1993c, 55-62.
- Mira, J. y col., «Local Learning in Networks of Universal Analogic Neurons», *Cybernetic & Systems: An International Journal*, Vol. 25, N. 2, marzo-abril, (1994), 259-273.
- Moreno-Díaz, R., «Deterministic and Probabilistic Neural Nets with Loops», *Mathematical Biosciences*, 11, (1971), 129-136.
- Moreno-Díaz, R. y McCulloch, W.S., «Circularities in Nets and the Concept of Functional Matrices», Proctor, L. (Ed.), *Byocybernetics of the CNS*, Little-Brown, Mass., 1968, 145-150.
- Moreno-Díaz, R. y Rubio, E., «A Theoretical Model for Layered Visual Processing», *Int. J. Bio-Med. Comp.*, 10, (1979), 134-143.
- Moreno-Díaz, R., Mira, J. y Roy, A., «Realización de Redes no Determinísticas», *Automática*, 12, (1972), 5-14.
- Mountcastle, V.B., «An Organizing Principle for Cerebral Function: the Unit Module and the Distributed System», Schmitt, F.O. y Worden, F.G. (Eds.), *The*

- Neuroscience Fourth Study Program*, The MIT Press, Cambridge, Mass., 1979, 21-42.
- Neumann, J. Von, «Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components», Shannon y McCarthy, (Eds.), *Automata Studies*, Princeton Univ. Press., Princeton, New Jersey, 1956.
- Nilsson, N.J., *Learning Machines*, McGraw-Hill, New York, 1965.
- Nordstrom, T. y Suensson, B., «Using and Designing Massively Parallel Computers for Artificial Neural Networks», *Journal of Parallel and Distributed Computing*, marzo, (1992), 260-285.
- Prieto, A. (Ed.), *Artificial Neural Networks. Lecture Notes in Computer Science*, 540, Springer-Verlag, Berlin, 1991.
- Ratliff, F., *Match Bands*, Holden-Day Inc., S. Francisco, 1965.
- Riseman, E.M. y Hanson, A.R., «A Methodology for the Development of General Knowledge-Based Vision Systems», Arbib, M.A. y Hanson, A.R. (Eds.), *Vision, Brain and Cooperative Computation*, MIT Press, Cambridge, Mass., 1987, 285-328.
- Rockmann, D. y Moraga, C., «Using Quadratic Perceptrons to Reduce Interconnection Density in Multilayer Neural Networks», en Prieto, A. (Ed.), *Artificial Neural Networks*, LNCS 540, Springer-Verlag, Heidelberg, 1991.
- Rosenblatt, F., *Principles of Neurodynamics*, Spartan Books, New York, 1959.
- Rovaris, E., *Imágenes, Muestreo y Computación Distribuida de Tipo Neuronal: Hacia un Enfoque Unificado*, Tesis Doctoral, E.T.S.I.T. Universidad de Las Palmas, 1992.
- Rumelhart, D.E. y col., «Learning Internal Representations by Error Propagation», en Rumelhart, D.E. y McClelland, J.L. (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, MIT Press, Mass., 1986.
- Sánchez-Sinécio, E. y Newcomb, R.W., «Neural Network Circuit Implementation», *Guest Editorial. IEEE Trans. on Neural Networks. Special Issue on Neural Network Hardware*, Vol. 4, N. 3, (1993), 385.
- Schmitt, F.O. y Worden, F.G., «The Neuroscience Fourth Study Program», The MIT Press, Mass., 1979.
- Segev, I., «Single Neurone Models: Oversimple, Complex and Reduced», *Trend in Neuroscience*, Vol. 15, N. 11, (1992), 414-421.

- Shannon, C.E. y McCarthy, J. (Eds.), *Automatas Studies*, Princeton University Press, N. Jersey, 1956.
- Shavlik, J.W. y Towell, G.G., «An Approach to Combining Explanation-based and Neural Learning Algorithms», *Connection Science*, Madison, WI, University of Wisconsin, Vol. 1, N. 3, (1989), 231-253.
- Steriade, M. y Llinás, R.R., «The Functional States of the Thalamus and the Associated Neuronal Interplay», *Physiological Reviews*, Vol. 68, N. 3, julio, (1988), 649-742.
- Thrun, S. y Mitchell, Tom M., «Explanation Based Learning: A Comparison of Symbolic and Neural Network Approaches», *Proceedings of the Tenth International Conference on Machine Learning*, Amherst, MA, (1993), 197-204.
- Treleaven, P., Pacheco, M. y Vellasco, M., «VLSI Architectures for Neural Networks», *IEEE Micro*, Vol. 9, N. 6, diciembre, (1989), 8-27.
- Vellasco, M.M.B.R. y Treleven, P.C., «The Generic Neuron Architectural Framework for the Automatic Generation», en Mira, J., Cabestany, J. y Prieto, A. (Eds.), *New Trends in Neural Computation*, LNCS 686, Springer-Verlag, Berlin, 1993, 476-481.
- Vidal, J.J., «Implementing Neural Nets with Programmable Logic», *IEEE Trans. on Acoustic, Speech and Signal Processing*, Vol. 36, N. 7, (1988), 1180-1190.
- Wawrzynek, J., Asanovic, K. y Morgan, N., «The Design of a Neuro-Microprocessor», *IEEE Trans. on Neural Networks. Special Issue on Neural Network Hardware*, Vol. 4, N. 3, (1993), 394-399.
- Wechsler, H. (Ed.), *Neural Networks for Perception* (Vol 1 y 2), Academic Press Inc, Boston, 1992.
- Werner, G. y Whitsel, B.L., «The Somatic Sensory Cortex: Functional Organization», Iggo, A. (Ed.), *The Somatosensory System (Handbook of Sensory Physiology, Vol. II)*, Springer-Verlag, Berlin, 1973.
- Widrow, B., «Generalization and Information Storage in Networks of Adaline Neurons», en Yovits, Jacobi y Goldstein, (Eds.), *Self-Organizing Systems*, Spartan Books, Washington, 1962, 345-461.
- XILINX, *Programmable Gate Arrays: XC4000 Logic Cell Array Family*, Technical Data Xilinx, San Jose, California, 1993.
- Yovits, M.C. y Cameron, S., *Self-Organizing Systems*, Pergamon Press, Oxford, 1960.