COMPUTACIÓN NEURONAL

MECANISMOS INSPIRADOS en el funcionamiento del cerebro

FRANCISCO CERVANTES PÉREZ, JOSÉ INCERA DIÉGUEZ Y SALVADOR MÁRMOL YAHYA

→A finales de los años 80, los sistemas que procesan información "al estilo del cerebro" definieron, junto con la inteligencia artificial, la Sexta Generación de Computadoras. Pero esta historia se inició en 1943, cuando el neurofisiólogo Warren McCulloch y el matemático Walter Pitts desarrollaron un modelo formal basado en una abstracción que integraba algunas de las propiedades básicas de los sistemas neurológicos biológicos, como el hecho de estar formada por unidades conectadas entre sí (neuronas o nodos). Ellos sugirieron que el cerebro de los seres vivos posee un gran poderío lógico y computacional. Posteriormente, este modelo simple de Redes de Neuronas Artificiales fue enriquecido por varios grupos de investigación al incorporar mecanismos de aprendizaje (por ejemplo la red llamada Perceptrón de Frank Rosenblatt, 1958, la cual era capaz de reconocer patrones sencillos y de generalizar similitudes entre patrones). Sin embargo, el campo de la computación neuronal obtuvo una gran aceptación y reconocimiento con la aparición del trabajo de los mapas autoorganizados de Kohonen (1982) y, sobre todo, del algoritmo de aprendizaje de Retropropagación del Error (Rumelhart et al, 1986).

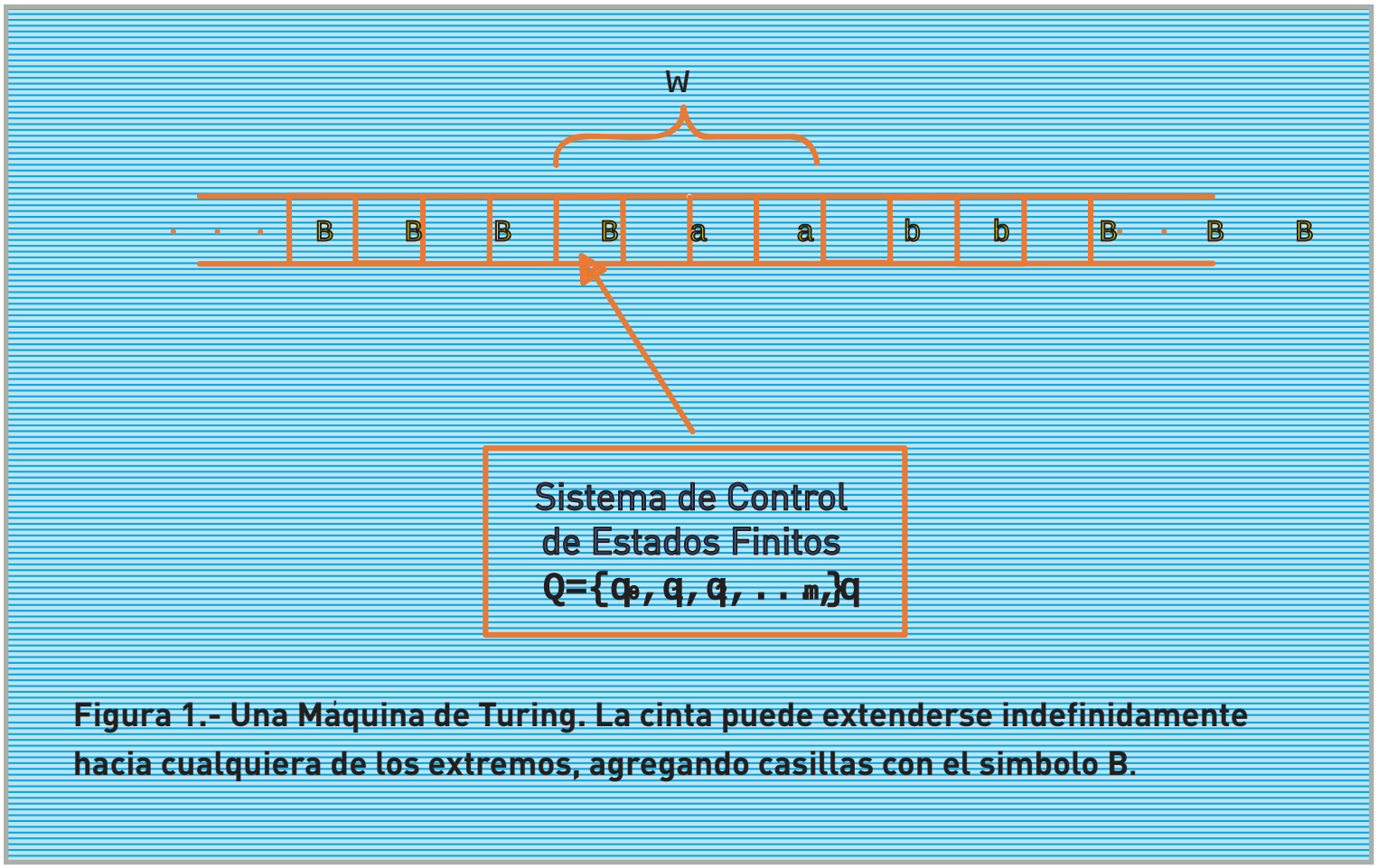


Figura 1.- Una Máquina de Turing. La cinta puede extenderse indefinidamente hacia cualquiera de los extremos, agregando casillas con el simbolo B.

→La máquina de Turing y la fisiología de lo computable.
La ciencia de la computación se originó en la década de 1930 con la máquina universal de Turing, desarrollada por el matemático inglés Alan Turing; La computación neuronal surgió durante la década siguiente, en 1943, con el trabajo de McCulloch y Pitts sobre un modelo formal de las propiedades básicas (anatómicas y fisiológicas) de la neurona biológica.

Una máquina de Turing M puede visualizarse como un sistema de Control de Estados Finitos (asociado a un conjunto finito de estados $Q= \{q_o,q_1,...,q_m\}$) conectado con una cinta de almacenamiento que puede extenderse en ambas direcciones de manera indefinida (ver figura 1).La cinta se divide en casillas que pueden contener un símbolo especial B (significa que el espacio está en blanco), o uno de cualquiera de los símbolos terminales $x_j, j=1,2,...,n,$ que pertenecen a un alfabeto X, o de los símbolos variables A_l ,$l=1,2,...,p$. El Control de Estados Finitos (CEF) está acoplado a la cinta a través de una cabeza de lectura y escritura, la cual se controla de acuerdo con unas Reglas de Transición,

$\S: Q \times (X \cup \{B\} \cup A_l) > Q \times (X \cup \{B\} \cup A_l) \times \{I,N,D\}$ que representan la forma en que una máquina M lleva a cabo un algoritmo específico. A cada instante, el CEF está en un estado q_i , se lee el símbolo en la casilla donde está colocada la cabeza lectora y, con base en la regla que aplique, el CEF pasa al estado q_j, se escribe un símbolo terminal, o variable, en la casilla y la cabeza de lectura/escritura se desplaza (a la Izquierda o a la Derecha) o no se mueve. Por ejemplo (ver figura 1), para determinar si una cadena $w=aabb$ pertenece, o no, al lenguaje $L=\{a^n b^n | n \geq o\}$, la máquina M se define como, $M = (Q,X,q_o,q_a,\S)$, donde:

$Q=\{q_o,q_1,q_2,q_3,q_a\}$ es un conjunto finito de estados,
$X = \{a,b\}$ es un alfabeto finito de símbolos terminales,
q_o es el estado inicial, con el que arranca el CEF todo proceso,
q_a es el estado de aceptación (solo si M está en q_a cuando termina de procesar la cadena bajo análisis se concluye que $w \in L$), y
$\S=\{(q_o,a,q_1,\&,D),(q_1,a,q_1,a,D),(q_1,b,q_2,\#,I),(q_2,a,q_2,a,I),$
$(q_2,\#,q_2,\#,I),(q_2,\&,q_o,\&,D),(q_1,\#,q_1,\#,D),$
$(q_o,\#,q_3,\#,D),(q_3,\#,q_3,\#,D),(q_3,B,q_a,B,N,)\}$

La máquina de Turing es la base de la Teoría de la Computabilidad que permitió el desarrollo de la computadora

digital, indicando claramente sus limitaciones: con ella solo se pueden programar funciones que sea realizables a través de un algoritmo.

Por otro lado, en el desarrollo del modelo de una neurona artificial, McCulloch y Pitts combinaron la neurofisiología con la lógica matemática, tomando en cuenta la propiedad de Todo-o-Nada de la activación de la neurona biológica para modelarla como una unidad binaria operando en una escala de tiempo discreto, $t=0,1,2,\ldots$. En los animales, las células del sistema nervioso central reciben señales en su cuerpo celular (soma) y arborización dendrítica, a través de sus conexiones con otras neuronas o con receptores asociados a órganos sensoriales (ver figura 2).

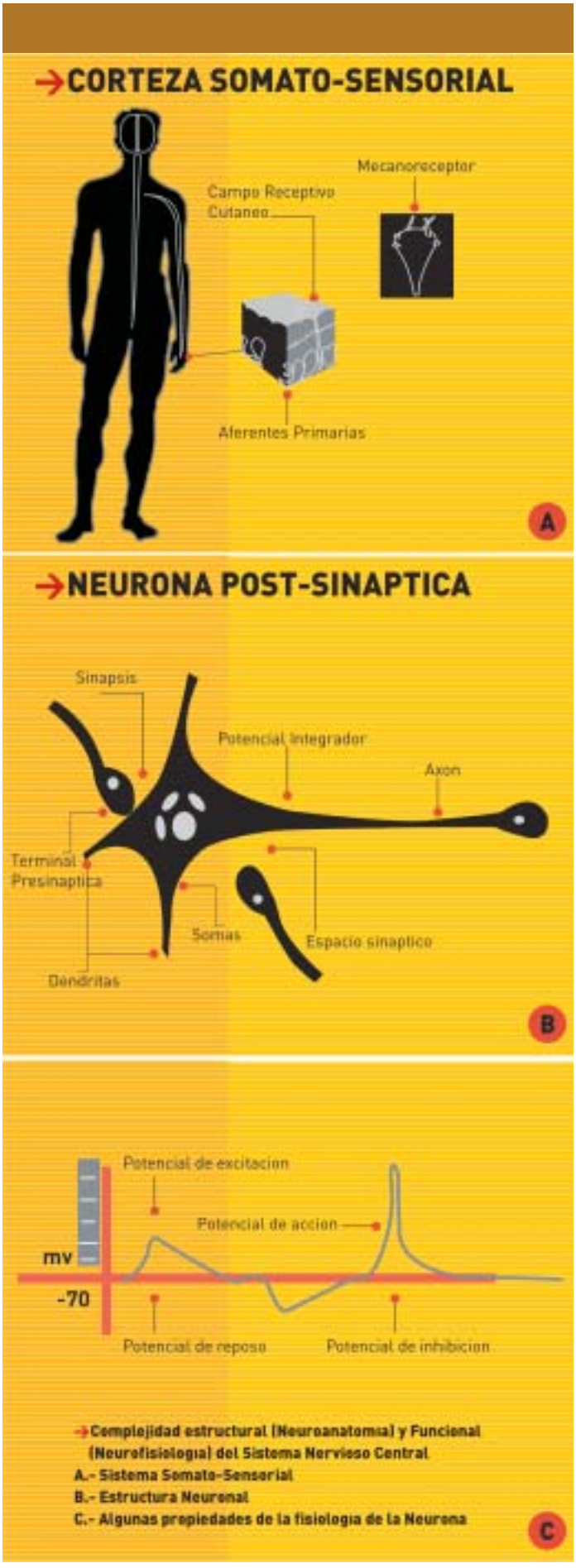
A cada instante, las entradas o salidas están activas o inactivas, lo que se indica con un valor binario de 1 o 0, respectivamente. Cada conexión, o sinapsis, de la salida de una neurona a la entrada de otra tiene asociado un nivel de ponderación (peso) Sea w_i el peso de la i-ésima entrada a una neurona dada, ésta se denomina como excitadora si $w_i>0$, o como inhibitoria si $w_i<0$. Cada neurona tiene asociado un valor de umbral $\emptyset$, de tal manera que cuando la suma ponderada de sus n entradas $\sum_{i=1}^{n} w_i x_i$ en el tiempo (t) es igual o mayor que $\emptyset$, la neurona "dispara" (asigna un valor de 1 a su salida en el axón) en el tiempo $t+1$, simulando la generación de un potencial de acción de la neurona biológica. Además, la neurona permanece inactiva (su salida tiene un valor de 0) en $t+1$ cuando no hay entradas activas, o cuando la suma ponderada de sus entradas en un tiempo t es menor que $\emptyset$. Esto es, si el valor de la i-ésima entrada es $x_i(t)$, entonces, la salida un tiempo después es:

$$y(t+1)=1 \quad si\ y\ solo\ si \sum_i w_i x_i(t) \geq \emptyset$$

Con este modelo simple de neurona artificial se pueden implementar las operaciones lógicas AND, OR y NOT (ver figura 3), lo que permite conformar una lógica booleana funcionalmente completa. Esto implica que, conectando de manera apropiada algunos de estos elementos, podemos formar Redes de Neuronas Artificiales (RNA) con un poderío computacional análogo al de los circuitos de control de una computadora digital. Así, en principio, con RNA se pueden realizar computaciones arbitrariamente complejas, y es por esta razón que el trabajo realizado por McCulloch y Pitts a menudo es considerado como el descubrimiento de la "Fisiología de lo Computable".

→Aprendizaje: "Programación Automá-tica de RNA"

En 1969, Minsky y Papert, en su libro Perceptrons, establecieron que el reto para el desarrollo de la computación neuronal consistía en programar (encontrar un conjunto de pesos y umbrales) automáticamente RNA de más de una capa de proceso (ver figura 4) para realizar funciones no lineales. Esto originó varios esfuerzos para



definir algoritmos que resolvieran este problema con base en alguna forma de aprendizaje, entre las que destacan el aprendizaje supervisado, el no supervisado, y el aprendizaje por reforzamiento.

En este texto sólo se presenta uno de los algoritmos de aprendizaje supervisado que han convertido las Redes Neuronales Artificiales en una valiosísima herramienta para la solución de problemas complejos de áreas como finanzas, control de consumos, comunicaciones, diagnóstico médico, meteorología, etc.

→Aprendizaje Supervisado: Algoritmo de Retropropagación del Error

Aprendizaje Supervisado es una forma de aprendizaje por ejemplos, donde el objetivo es identificar un mapeo entre p parejas de entradas y salidas conocidas, $(x_i,y_i), i=1,2,...,p$, durante un proceso de entrenamiento. El primer esquema fue propuesto por Hebb en 1949: fortalecer los pesos asociados a sinapsis cuya actividad coincida con la activación de la neurona postsináptica. Posteriormente, a finales de los años 1950 y en los 1960, Rosenblatt propuso el algoritmo de aprendizaje denominado Perceptrón: incrementar los pesos asociados con entradas activas si la neurona postsináptica no dispara cuando debería haberlo hecho, y disminuirlos cuando ésta dispare y no debería hacerlo.

Ambos algoritmos sirvieron de inspiración para muchos de los algoritmos actuales de aprendizaje en sistemas de RNA. El algoritmo de Retropropagación del Error

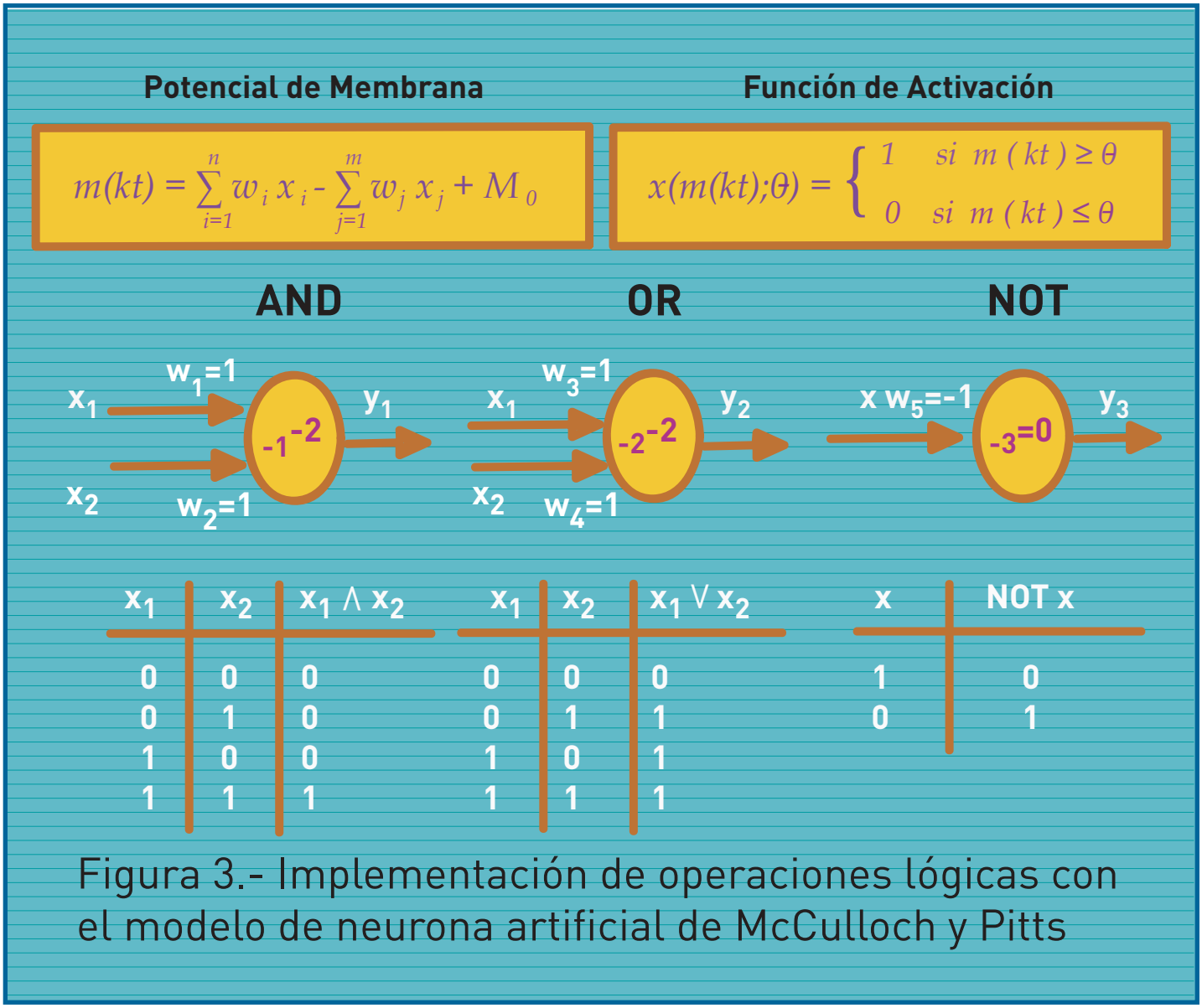


Figura 3.- Implementación de operaciones lógicas con el modelo de neurona artificial de McCulloch y Pitts

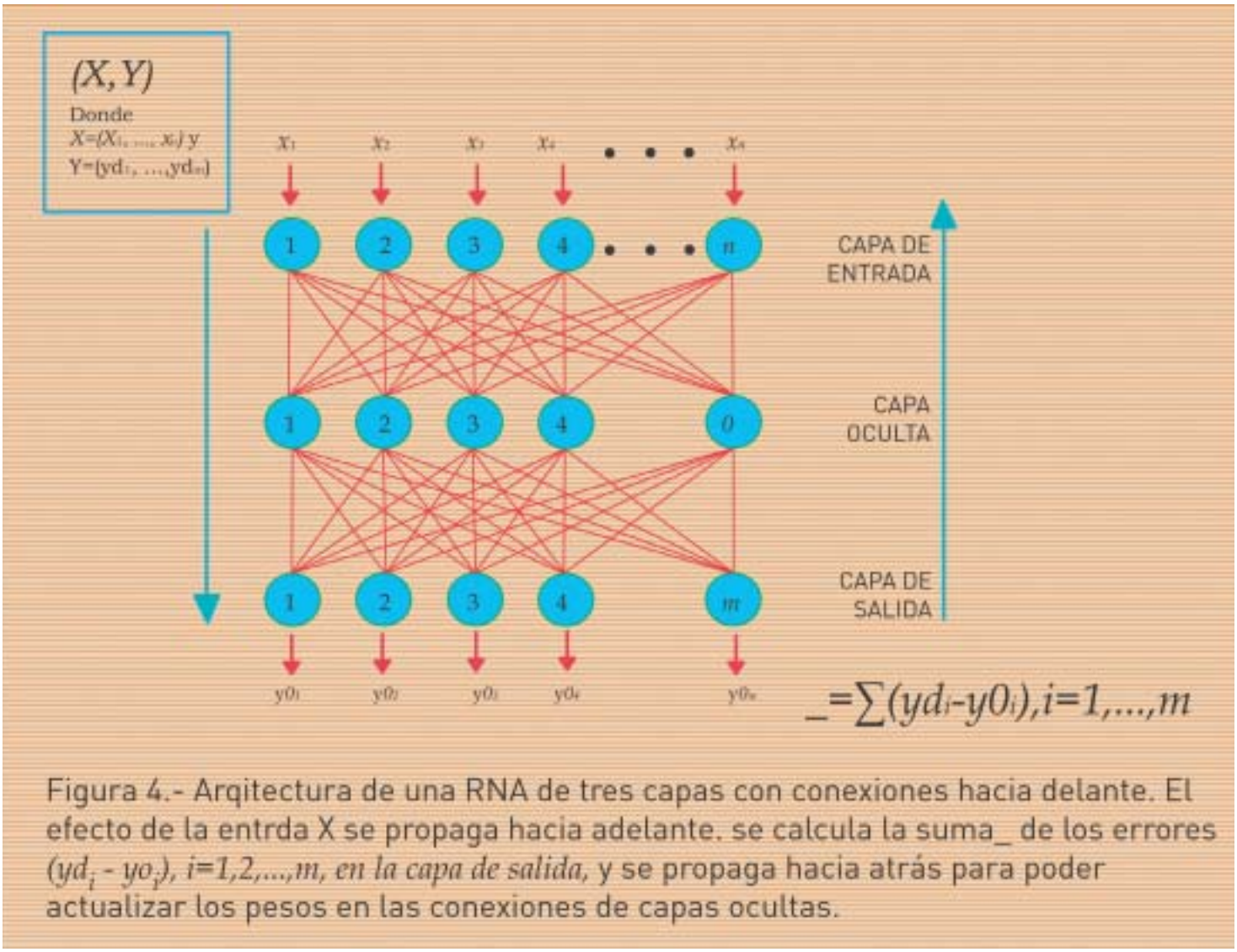


Figura 4.- Arqitectura de una RNA de tres capas con conexiones hacia delante. El efecto de la entrda X se propaga hacia adelante. se calcula la suma_ de los errores $(yd_i - yo_i)$, $i=1,2,...,m$, *en la capa de salida,* y se propaga hacia atrás para poder actualizar los pesos en las conexiones de capas ocultas.

es el que más aplicaciones tiene en la industria, en los negocios y en la academia, el cual fue propuesto por Rumelhart, Hinton y Williams en 1986.

En este método, la base de ejemplos conocidos se divide en dos conjuntos: uno de entrenamiento y otro de prueba. El primero se usa para identificar el mapeo entre los patrones de entrada y salida, mientras que el segundo sirve para validar el desempeño de la RNA ante casos nuevos. Sea $X_1=(x_{11},x_{12},...,x_{1n})$ el vector de entrada del primer patrón del conjunto de entrenamiento y $Y_1=(y_{11},y_{12},...,y_{1m})$ el vector de salidas correspondiente. Al presentar X_1 en la entrada de la RNA y propagar su efecto hasta la Capa de Salida, se genera la salida o_{1k},$k=1,2,...,m$. Al compararla con los valores y_{1k}, se genera una función de error $E_1=\sum_k \delta^2_{1k}$, donde $\delta_{1k}=(y_{1k}-o_{1k})$ Es el error en la neurona *k* de la Capa de Salida. El objetivo del proceso de aprendizaje es minimizar este error, modificando los pesos de las conexiones entre las distintas neuronas de la red. Para esto se utiliza la regla del gradiente descendente:

$$\Delta w_{kj}=-\partial E_1/\partial w_{kj}=2\sum_k(y_{1k}-o_{1k})\partial o_{1k}/\partial w_{kj}$$

Los pesos se cambian aplicando esta fórmula a los pesos asociados a las neuronas de la capa de salida; sin embargo, para aplicar la misma regla con Δw_{ji}, al cambiar los pesos entre elementos de la Capa de Entrada y la neurona *j* de la Capa Oculta, se usa un error estimado $\delta'_{1j}=(\sum_k \delta 1_k w_{kj})f'_{1j}$. Esto es, se propaga hacia atrás el error obtenido en las neuronas de la Capa de Salida a través de la conexión que reciben de la neurona *j* de la capa oculta.

Cuando este procedimiento se repite con todas las parejas (X_i,Y_i) del conjunto de entrenamiento se dice que hemos completado una "Época". El número de épocas

durante un proceso de entrenamiento es variable, pues está ligado al momento en que se minimiza el error asociado al conjunto de entrenamiento. Por otro lado, aunque el objetivo es minimizar la función de error asociada al conjunto de entrenamiento, periódicamente se presenta a la red el conjunto de prueba. La meta principal es construir una RNA con el conjunto de pesos y umbrales que minimice la función de error asociada con este segundo conjunto.

→Sistemas que procesan información al "estilo del cerebro"

Los sistemas de información donde se utilizan tecnologías derivadas de la computación neuronal son considerados sistemas que procesan información al "estilo del cerebro", los cuales se han aplicado en la solución de problemas complejos en clasificación de patrones y en aproximación de curvas altamente no lineales para tareas de pronóstico, entre otros.

Este sistema para reconocer patrones tiene múltiples aplicaciones, entre otras, en la industria bancaria. Por ejemplo, en los sistemas para detectar fraudes en transacciones con tarjeta de crédito se usan RNA para modelar los patrones de consumo de los tarjeta-habientes. Estos modelos sirven como punto de referencia para calificar transacciones fraudulentas, mismas que se han incrementado a raíz del comercio más generalizado a través de Internet. Adicionalmente, este método también se ha utilizado con éxito en la configuración y administración de portafolios de inversiones, donde las RNA logran mejores predicciones que las técnicas tradicionales, tanto en el pronóstico de los valores de acciones como de indicadores financieros.

Aunque hemos discutido sobre las características computacionales de los modelos de RNA, debe señalarse que para resolver problemas complejos donde las técnicas tradicionales producen resultados pobres, o no ofrecen solución alguna, a veces es necesario integrarlas con otro tipo de tecnologías originadas en la inteligencia artificial o en las matemáticas. Esto es, el reto actual es construir "agentes inteligentes" híbridos, basados en la integración de RNA con sistemas expertos, algoritmos genéticos, métodos estadísticos o modelos matemáticos. Por ejemplo, en las Redes de Comunicaciones de Paquetes, como internet, la predicción en línea del tráfico en una Red de Área Local (*LAN, por sus siglas en inglés*) es un tema de gran importancia ya que, entre otras cosas, permitiría mejorar significativamente la calidad de servicio asociada a aplicaciones multimedia en tiempo real, como telefonía IP o videoconferencias. Nuestro grupo de investigación está desarrollando una arquitectura de multiagentes basados en RNA, colocados en diferentes ruteadores, para predecir en tiempo real el tráfico global de la LAN. Además, se modela el desempeño de la RNA con un método estadístico, el cual se combina con una ecuación matemática asociada a las características del ruteador de salida de la LAN, para determinar el instante en que el ancho de banda reservado es insuficiente. La predicción de tráfico con RNA también podría utilizarse en algoritmos de control para evitar congestión en redes de computadoras y en los esquemas de administración de las LAN.

Francisco Cervantes Pérez Es ingeniero mecánico electricista y maestro en ingeniería eléctrica por la Facultad de Ingeniería de la UNAM, además de doctor en ciencias de la computación y de la información por la Universidad de Massachusetts, EUA. Sus líneas de investigación son: el análisis de las propiedades dinámicas y computacionales de los sistemas neuronales que forman el sustrato de los procesos de coordinación sensorio-motora en seres vivos, y la síntesis de modelos computacionales basados en la Teoría de Esquemas y de redes neuronales biológicas para construir autómatas que permitan resolver problemas prácticos en robótica, control no-lineal y reconocimiento de patrones.

José Alberto Inciera Diéguez Es ingeniero en electrónica con especialidad en sistemas digitales por la Universidad Autónoma Metropolitana (UAM), maestro en ciencias de la computación por el Colegio Imperial de Ciencia, Tecnología y Medicina de la Universidad de Londres, Inglaterra, y doctor en informática por la Universidad de Rennes 1, en Francia. Ha sido docente en la UAM, la UNAM, y el ITAM. Sus áreas de investigación son redes de comunicación, evaluación de prestaciones y calidad de servicio; campo en el que ha publicado varios artículos, además de participar en foros nacionales e internacionales, así como en diversos proyectos de consultoría.

Juan Salvador Mármol Yahya Es ingeniero en computación por el Instituto Tecnológico Autónomo de México, maestro en ingeniería por la UNAM (División de Estudios de Postgrado de la Facultad de Ingeniería) y candidato a doctor por la Universidad del Sur de California, además de maestro en ciencias por la misma universidad. Actualmente es asistente de investigación en el Brain Simulation Lab de la Universidad del Sur de California donde desarrolla varias herramientas neuroinformáticas y modelos computacionales del cerebro.