Redes neuronales artificiales: aprendizaje supervisado

Redes neuronales: aspectos generales

Al intentarse construir máquinas inteligentes surgió, naturalmente, la mente humana como modelo. Por lo tanto, dentro de la inteligencia artificial resultó obvia la idea de simular directamente el funcionamiento del cerebro en una computadora, lo que justificó el interés por las arquitecturas basadas en redes neuronales. Estos modelos reciben el nombre de conexionistas.

Estas iniciativas referidas a las redes neuronales estuvieron inspiradas en hechos ya conocidos sobre el funcionamiento del cerebro. Entre los precursores del estudio anatómico y neurológico del cerebro se destaca Ramón y Cajal (1900), quién identificó su estructura en red. Merece también destacarse la contribución de Charles Sherring, en la misma época, por identificar la presencia de sinapsis en los vínculos entre diferentes células o neuronas.

Como se recordará, fueron Warren McCullock y Walter Pitts (1943) quienes desarrollaron un primer modelo formal destinado a representar el comportamiento de las neuronas. Es decir, el primer modelo neuronal artificial. Sin embargo, este primer modelo neuronal carecía de la capacidad de aprendizaje, una de las principales características de los cerebros de los animales y del hombre.

Siete años después, Alan Turing (1950) se anticipó al nacimiento de la Inteligencia Artificial (1956) al plantear la inquietud sobre la posibilidad que las máquinas piensen, en su famoso artículo *Computing Machinery and Intelligence* en la revista MIND. Allí propuso el test que lleva su nombre y los principios del aprendizaje automático. También analizó otros aspectos relativos a las capacidades de las máquinas.

Poco después, Ross Ashby publicó su libro titulado *Design for a Brain* (1952), donde presentó los resultados de numerosos experimentos y algunas ideas esenciales que posibilitaron más tarde el desarrollo de redes neuronales artificiales con capacidad de aprendizaje.

Frank Rosenblatt, psicólogo estadounidense, estuvo al frente de varias investigaciones importantes, presentando en una de ellas el desarrollo del Perceptrón (1958), un clasificador binario o discriminador lineal capaz de generar una predicción basándose en un algoritmo que combina los pesos de sus entradas. Para ello, se apoyó en los descubrimientos biológicos de Ramón y Cajal, a los que incorporó las ideas propuestas por MaCulloch y Pitts en el primer modelo neuronal artificial.

En 1969 Minsky y Papert demostraron las severas limitaciones del perceptrón, en ese entonces, el modelo neuronal por excelencia, provocando una gran pérdida de confianza en este campo y desalentando tanto la investigación como su respaldo económico. Ésta recién sería recuperada a principios de los ’80 a partir de las propuestas novedosas de John Hopfield y otros investigadores.

CONTINUAR

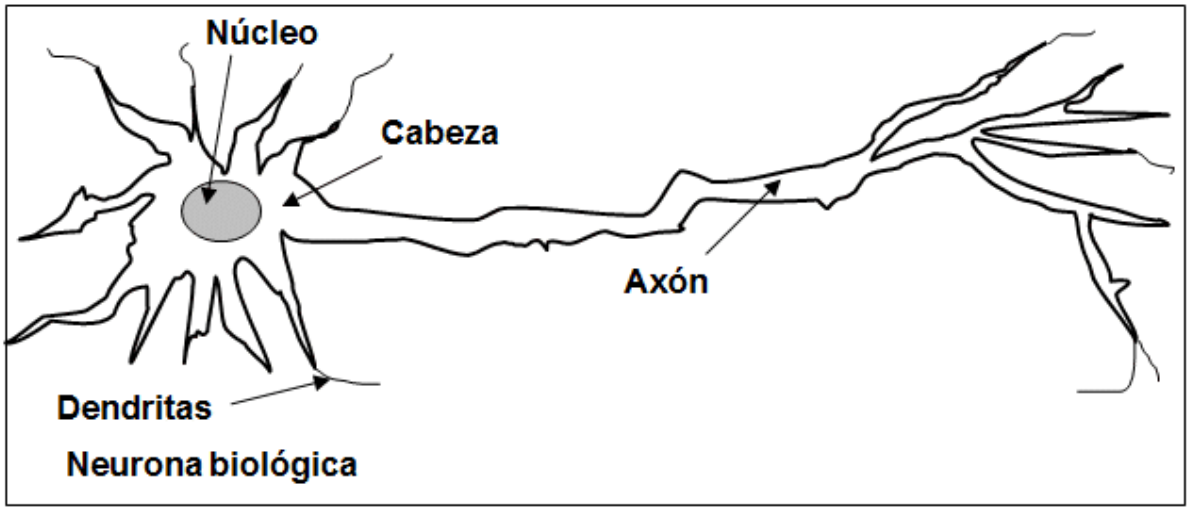
Redes neuronales: fundamentos biológicos

El estudio anatómico detallado del cerebro reveló que un ser humano tiene en el orden de 100.000 millones de neuronas con una conectividad media de 10.000. Los elementos principales en una neurona simplificada son: cabeza, núcleo, axón y dendritas.

> “Las dendritas… proporcionan área superficial en abundancia para facilitar la conexión con los axones de otras células. Los axones influyen sobre las dendritas a través de espacios estrechos, una superficie de contacto, conocida como sinapsis. La estimulación de algunas sinapsis hace que las neuronas se disparen al regular la intensidad y naturaleza de la influencia de una neurona sobre otra”. (Red neuronal, s.f., https://bit.ly/3v5DN6z).

Esta acción puede ser de naturaleza excitadora o inhibidora.

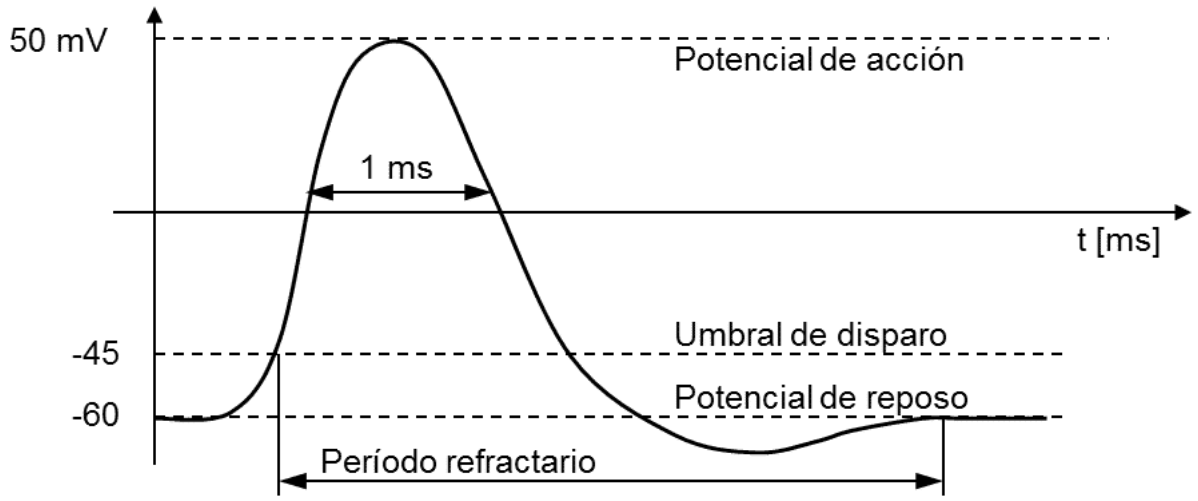
Figura 1: Esquema simplificado de una neurona



Fuente: elaboración propia.

Una neurona no hace nada o produce una salida de potencia completa, por lo que se la reconoce como un dispositivo de todo o nada. Se manifiesta como un impulso eléctrico que se desplaza por el cuerpo de la célula, por el axón hacia sus ramificaciones y se dice, entonces, que la neurona se ha disparado.

Figura 2: Pulso de disparo de una neurona



Fuente: [Imagen sin título sobre pulso de disparo de una neurona]. (s. f.).

Las señales nerviosas son eléctricas en el interior de las neuronas y químicas en su exterior. En este último caso, se basan en el intercambio de neurotransmisores, determinado por la membrana neuronal y los iones presentes a ambos lados de ella. La membrana contiene canales iónicos selectivos al tipo de ion, algunos son pasivos (simples poros en la membrana) y otros activos (poros de apertura selectiva). Los iones son de sodio (Na), potasio (K), calcio (Ca) y proteínas, y cada uno presenta distinta permeabilidad. “En estado de reposo el interior de la neurona permanece cargado negativamente en relación al medio exterior, existiendo una diferencia de potencial de -60 mV” (Hernández Paxtián, 2011, https://bit.ly/3tCPvFl). Cuando se hayan producido suficientes excitaciones, el potencial de la neurona puede elevarse por encima de los -45 mV que es el umbral de disparo. En ese momento se abren bruscamente los canales de sodio, de modo que esos iones (Na+), cuya concentración exterior es alta,

ingresan masivamente al interior, provocando una despolarización de la neurona que pasa de -60 mV a +50 mV, volviendo luego a su condición de reposo de -60 mV. Después del disparo, la neurona entra en un período refractario hasta quedar habilitada para un próximo ciclo. Debe observarse que el pulso es digital, se produce al superar el umbral y todos son de la misma magnitud. Las frecuencias de disparo están entre 1 y 100 pulsos por segundo.

El desempeño de una sinapsis no corresponde a un valor predeterminado, por el contrario, es variable. Esta plasticidad sináptica es la que determina la capacidad de aprendizaje, que está ampliamente respaldada por evidencias experimentales. A la plasticidad sináptica se suman otras formas de modelado, representado por el establecimiento de nuevas conexiones, eliminación de otras e inclusive muerte neuronal.

CONTINUAR

Redes neuronales artificiales: estructuras y clasificaciones

Al igual que en los sistemas biológicos, los elementos básicos de los modelos neuronales artificiales son las unidades activas, que se agrupan en conjuntos muy numerosos organizadas en capas, constituyendo un sistema con funcionamiento propio. La información de entrada atraviesa la red neuronal y cada unidad es sometida a operaciones en las que intervienen los valores de entrada y los pesos sinápticos, obteniéndose, así, los valores de salida. Previamente a la salida de cada unidad, hay una función limitadora que, operando con un umbral, modifica el resultado, imponiendo un límite sobre el valor de salida antes de propagarse a otra neurona. Esta función se conoce como función de activación.

En resumen, un sistema neuronal artificial o conexionista está compuesto por los siguientes cinco elementos:

- un conjunto de unidades neuronales o procesadores elementales;
- un patrón de conectividad o arquitectura;
- una dinámica de activaciones;

- un modo de aprendizaje;
- un entorno de operación.

Las **unidades neuronales** son dispositivos de cálculo muy simples que operan a partir de los valores de un vector de entrada, los pesos sinápticos, la función de activación y un umbral para determinar el valor de cada salida. Las funciones de activación definen si los valores de las salidas son discretas o continuas.

La **arquitectura** de la red define su topología, es decir la forma en que las unidades neuronales son conectadas a sus entradas y la salida a través de los pesos sinápticos. Las conexiones son direccionales, es decir, la información se propaga en un sentido y las unidades se distribuyen en estructuras de agrupamiento denominadas capas. Dentro de cada capa, las neuronas pueden, a su vez, agruparse por afinidad formando clústeres. Al definir la arquitectura también se asignan los tipos de funciones de activación a las unidades de cada capa o de cada clúster.

La **dinámica de activación** determina la forma en que las neuronas de una red actualizan sus estados: con dinámica síncrona o dinámica asíncrona. En el primer caso, los estados se actualizan en función de un reloj común, realizándose el proceso por capas, desde la entrada a la salida. En el segundo caso, cada unidad actualiza su estado en forma independiente, sin considerar cuando lo hacen las demás.

El **modo de aprendizaje** o entrenamiento establece la forma en que se le asignarán sus valores a los pesos sinápticos de los vínculos entre las unidades neuronales. Como ya fue dicho, esta plasticidad sináptica es la que define el desempeño de la red ante cierta condición de entrada. Se reconocen los siguientes cuatro modos de aprendizaje. Los primeros dos son considerados básicos:

1. **Supervisado:** se presenta a la red un conjunto de patrones de entrada y sus correspondientes salidas, ajustándose los pesos sinápticos hasta que el desempeño de la red es el deseado.

2. **No supervisado o autoorganizado:** solo se presenta a la red el conjunto de patrones de entrada y la red en el proceso de aprendizaje identifica regularidades, realiza agrupamientos (*clustering*) o estabiliza su salida en torno a cierta condición de entrada.

3. **Híbrido:** se reúnen en una misma red los dos modos básicos de aprendizaje, los que son aplicados normalmente en diferentes capas de unidades.

4. **Reforzado:** se encuentra en un punto intermedio entre los dos modos básicos, y el valor de referencia durante el proceso no corresponde a la salida deseada, sino a un índice de rendimiento de la red.

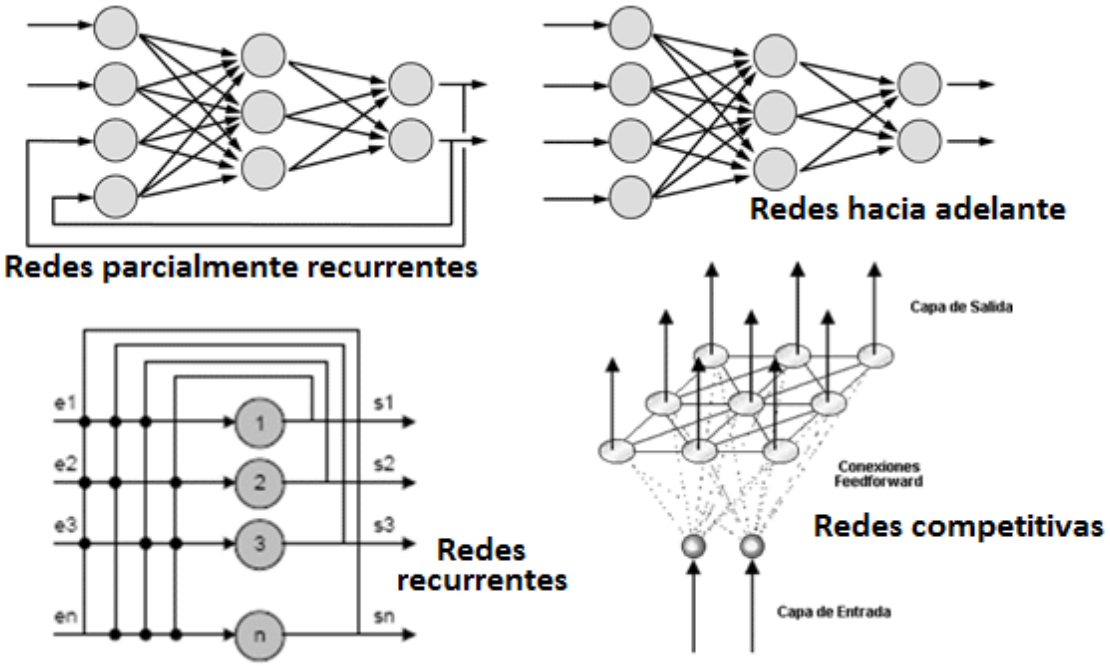
El **entorno de operación** es considerado un elemento propio de la red debido a la alta especialización de estos modelos. Es decir, la red es específicamente diseñada y entrenada para desempeñar cierta actividad en un entorno de operación muy preciso, por lo que no es trasladable de un escenario a otro. Es trasladable la experiencia en cuanto a la conveniencia de determinada arquitectura para resolver ciertos problemas, pero en modo alguno a la propia red.

En la figura 3 se presenta un esquema con la clasificación de los principales modelos neuronales según el tipo de aprendizaje y en la figura 4 se muestran las principales arquitecturas.

Figura 3: Clasificación según modo de aprendizaje

Fuente: elaboración propia.

Figura 4: Arquitecturas típicas de los principales modelos neuronales



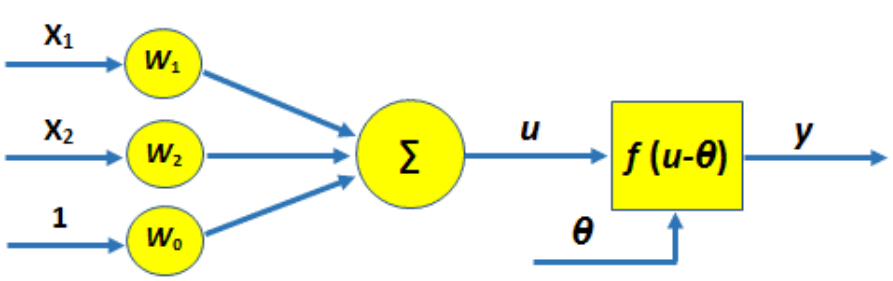
Fuente: [Imagen sin título sobre arquitecturas de los nodos neuronales]. (s.f.).

CONTINUAR

Redes neuronales simples de una capa: aprendizaje supervisado

Tal como se anticipó, en la arquitectura de las redes se reconoce la disposición de las unidades en capas y naturalmente los modelos más sencillos son los de una capa. El modelo más simple posible es el de una sola unidad y, obviamente, una capa. Aquí se aprovechará para entrar en detalle con respecto a la función de activación, todavía pendiente de un tratamiento más detallado. Se toma como ejemplo una unidad neuronal (perceptrón) con dos entradas, destinada a reproducir el comportamiento de una puerta "OR". La pequeña red y las condiciones a ser reproducidas se muestran en la figura 5.

Figura 5: Modelo de un perceptrón que representa una compuerta "OR"

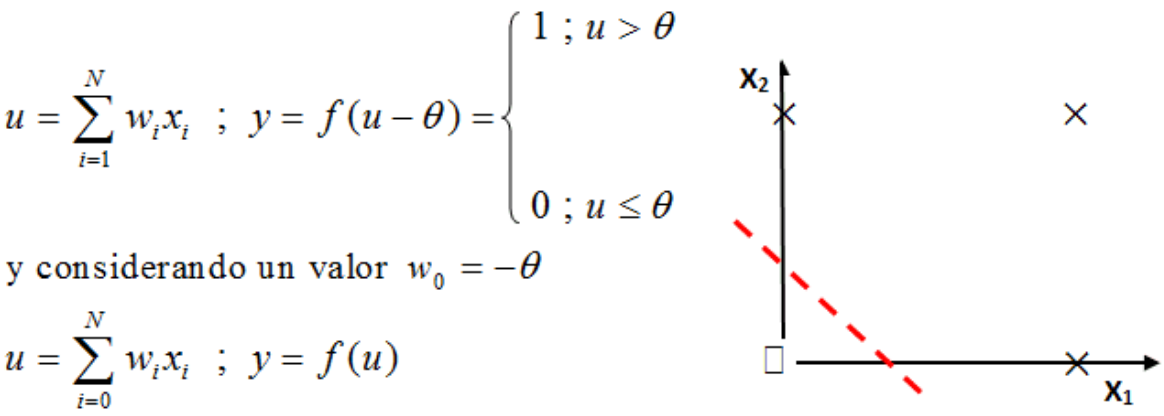


OR

X_1	X_2	S
0	0	0
0	1	1
1	0	1
1	1	1

Fuente: elaboración propia.

Figura 6: Interpretación de la compuerta



Fuente: [Imagen sin título sobre interpretación de la compuerta]. (s.f.).

Obsérvese que en lugar de ingresarse el valor de θ se incorpora un peso sináptico adicional (**W_0**) que multiplica a una entrada de referencia (1). La finalidad es ajustar el valor del escalón conjuntamente con los pesos sinápticos, simplificándose el algoritmo.

El entrenamiento del perceptrón para que se desempeñe como una compuerta “OR” implica los siguientes pasos:

$$\{w\} = \begin{Bmatrix} 1,5 \\ 0,5 \\ 1,5 \end{Bmatrix}$$

1) Proponer un conjunto de valores iniciales aleatorios para el vector de entrada.

2) Ajustar los pesos sinápticos de acuerdo a la expresión: $\{w\}_{t+1} = \{w\}_t + \alpha[s - y_t]\{x\}$

3) Repetir el proceso de ajuste ordenadamente para los cuatro conjuntos de "entrada – salida" y repetir esta secuencia de ajuste hasta que los errores sean nulos.

Debe observarse que los valores de la salida "***y***" es siempre binaria, conforme a la función de activación utilizada ($\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{u} - \boldsymbol{\theta}) = [0, 1]$).

Además, el coeficiente **α** presente en la expresión de ajustes de pesos es siempre "1". Se denomina ***factor de aprendizaje*** y es utilizada en salidas continuas.

A continuación, se presentan los ciclos de ajustes de los pesos en tablas para facilitar su interpretación:

Tabla 1: Ciclos de ajuste

Primer ciclo de ajuste

(x , s)		Pesos	Salida calculada	Ajuste de pesos	Error
Entrada	Salida				
1 0 0	0	1,5 0,5 1,5	f (1,5) = 1	$\begin{Bmatrix}1,5\\0,5\\1,5\end{Bmatrix} + (-1)\begin{Bmatrix}1\\0\\0\end{Bmatrix} = \begin{Bmatrix}0,5\\0,5\\1,5\end{Bmatrix}$	-1
1 0 1	1	0,5 0,5 1,5	f (2,0) = 1	$\begin{Bmatrix}0,5\\0,5\\1,5\end{Bmatrix} + (0)\begin{Bmatrix}1\\0\\1\end{Bmatrix} = \begin{Bmatrix}0,5\\0,5\\1,5\end{Bmatrix}$	0
1 1 0	1	0,5 0,5 1,5	f (1,0) = 1	$\begin{Bmatrix}0,5\\0,5\\1,5\end{Bmatrix} + (0)\begin{Bmatrix}1\\1\\0\end{Bmatrix} = \begin{Bmatrix}0,5\\0,5\\1,5\end{Bmatrix}$	0
1 1 1	1	0,5 0,5 1,5	f (2,5) = 1	$\begin{Bmatrix}0,5\\0,5\\1,5\end{Bmatrix} + (0)\begin{Bmatrix}1\\1\\1\end{Bmatrix} = \begin{Bmatrix}0,5\\0,5\\1,5\end{Bmatrix}$	0

Segundo ciclo de ajuste

(x , s)		Pesos	Salida calculada	Ajuste de pesos	Error
Entrada	Salida				
1 0 0	0	0,5 0,5 1,5	f (0,5) = 1	$\begin{Bmatrix}0,5\\0,5\\1,5\end{Bmatrix} + (-1)\begin{Bmatrix}1\\0\\0\end{Bmatrix} = \begin{Bmatrix}-0,5\\0,5\\1,5\end{Bmatrix}$	-1
1 0 1	1	-0,5 0,5 1,5	f (1,0) = 1	$\begin{Bmatrix}-0,5\\0,5\\1,5\end{Bmatrix} + (0)\begin{Bmatrix}1\\0\\1\end{Bmatrix} = \begin{Bmatrix}-0,5\\0,5\\1,5\end{Bmatrix}$	0
1 1 0	1	-0,5 0,5 1,5	f (0,0) = 0	$\begin{Bmatrix}-0,5\\0,5\\1,5\end{Bmatrix} + (1)\begin{Bmatrix}1\\1\\0\end{Bmatrix} = \begin{Bmatrix}0,5\\1,5\\1,5\end{Bmatrix}$	1
1 1 1	1	0,5 1,5 1,5	f (3,5) = 1	$\begin{Bmatrix}0,5\\1,5\\1,5\end{Bmatrix} + (0)\begin{Bmatrix}1\\1\\1\end{Bmatrix} = \begin{Bmatrix}0,5\\1,5\\1,5\end{Bmatrix}$	0

Tercer ciclo de ajuste

(x , s)		Pesos	Salida calculada	Ajuste de pesos	Error
Entrada	Salida				
1 0 0	0	0,5 1,5 1,5	f (0,5) = 1	$\begin{Bmatrix}0,5\\1,5\\1,5\end{Bmatrix} + (-1)\begin{Bmatrix}1\\0\\0\end{Bmatrix} = \begin{Bmatrix}-0,5\\1,5\\1,5\end{Bmatrix}$	-1
1 0 1	1	-0,5 1,5 1,5	f (1,0) = 1	$\begin{Bmatrix}-0,5\\1,5\\1,5\end{Bmatrix} + (0)\begin{Bmatrix}1\\0\\1\end{Bmatrix} = \begin{Bmatrix}-0,5\\1,5\\1,5\end{Bmatrix}$	0
1 1 0	1	-0,5 1,5 1,5	f (1,0) = 1	$\begin{Bmatrix}-0,5\\1,5\\1,5\end{Bmatrix} + (0)\begin{Bmatrix}1\\1\\0\end{Bmatrix} = \begin{Bmatrix}-0,5\\1,5\\1,5\end{Bmatrix}$	0
1 1 1	1	-0,5 1,5 1,5	f (2,5) = 1	$\begin{Bmatrix}-0,5\\1,5\\1,5\end{Bmatrix} + (0)\begin{Bmatrix}1\\1\\1\end{Bmatrix} = \begin{Bmatrix}-0,5\\1,5\\1,5\end{Bmatrix}$	0

Fuente: [Imagen sin título sobre Ciclos de ajuste]. (s.f.).

Un cuarto ciclo de ajuste demuestra que las cuatro entradas dan por resultado un error nulo, dejándose al lector su comprobación. Esto significa que, con el ajuste realizado en los pesos sinápticos, el perceptrón tiene el mismo desempeño que una compuerta “OR”.

Obtenidos los pesos, el próximo paso es la interpretación de los resultados.

A partir de la interpretación gráfica del comportamiento de la compuerta, puede escribirse:

$$W_0 + W_1 X_1 + W_2 X_2 = 0$$

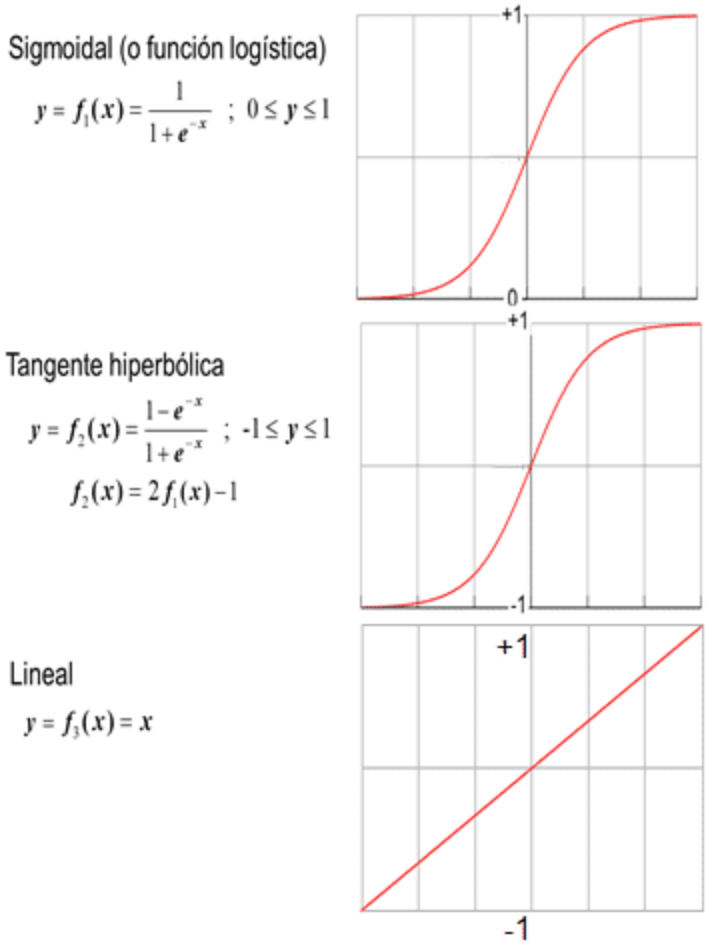
Es decir que: $X_2 = - W_0 / W_2 - (W_1 / W_2) X_1$

Considerando los pesos sinápticos: $W_0 = -0,5$, $W_0 = 1,5$, $W_0 = 1,5$

Se obtiene la expresión de la función discriminante: $X_2 = 0,5 / 1,5 - (1,5/ 1,5) X_1 = 0,33 - X_1$

La función de activación utilizada es binaria, acorde al tipo de problema. Puede ocurrir que sea necesaria una función también binaria, pero de evolución suave entre los dos valores extremos (derivable) o una función continua ya que la salida del modelo representará valores reales. En el primer caso se trata de una unidad perceptrón y en el segundo de una unidad *adaline* (*adaptative linear element)*. Las funciones de activación más usadas son:

Figura 7: Funciones de activación



Fuente: [Imagen sin título sobre Funciones de activación]. (s.f.).

Con el fin de completar la presentación del perceptrón operando individualmente, se recurre a un caso muy simple pero efectivo y muy útil, que se presenta a continuación.

El problema se refiere a la necesidad de reconocer el valor que representan las teclas presionadas en un "*pad* numérico" a partir de considerar en cada caso la fila y columna a las que pertenecen. Es el caso típico de

reconocimiento de teclas en todo teclado. En la figura 8 se presenta el esquema de este caso y un detalle de los valores que adoptan las señales de entrada en cada caso.

Figura 8: Detalle de señales de entrada originadas en el teclado

Entradas correspondientes a líneas de scan

1	X_1	X_2	X_3	X_4	X_5	X_6	X_7	Salida
1	0	0	0	1	0	1	0	0
1	1	0	0	0	0	0	1	1
1	1	0	0	0	0	1	0	2
1	1	0	0	0	1	0	0	3
1	0	1	0	0	0	0	1	4
1	0	1	0	0	0	1	0	5
1	0	1	0	0	1	0	0	6
1	0	0	1	0	0	0	1	7
1	0	0	1	0	0	1	0	8
1	0	0	1	0	1	0	0	9

Fuente: [Imagen sin título sobre señales originadas en el teclado]. (s.f.).

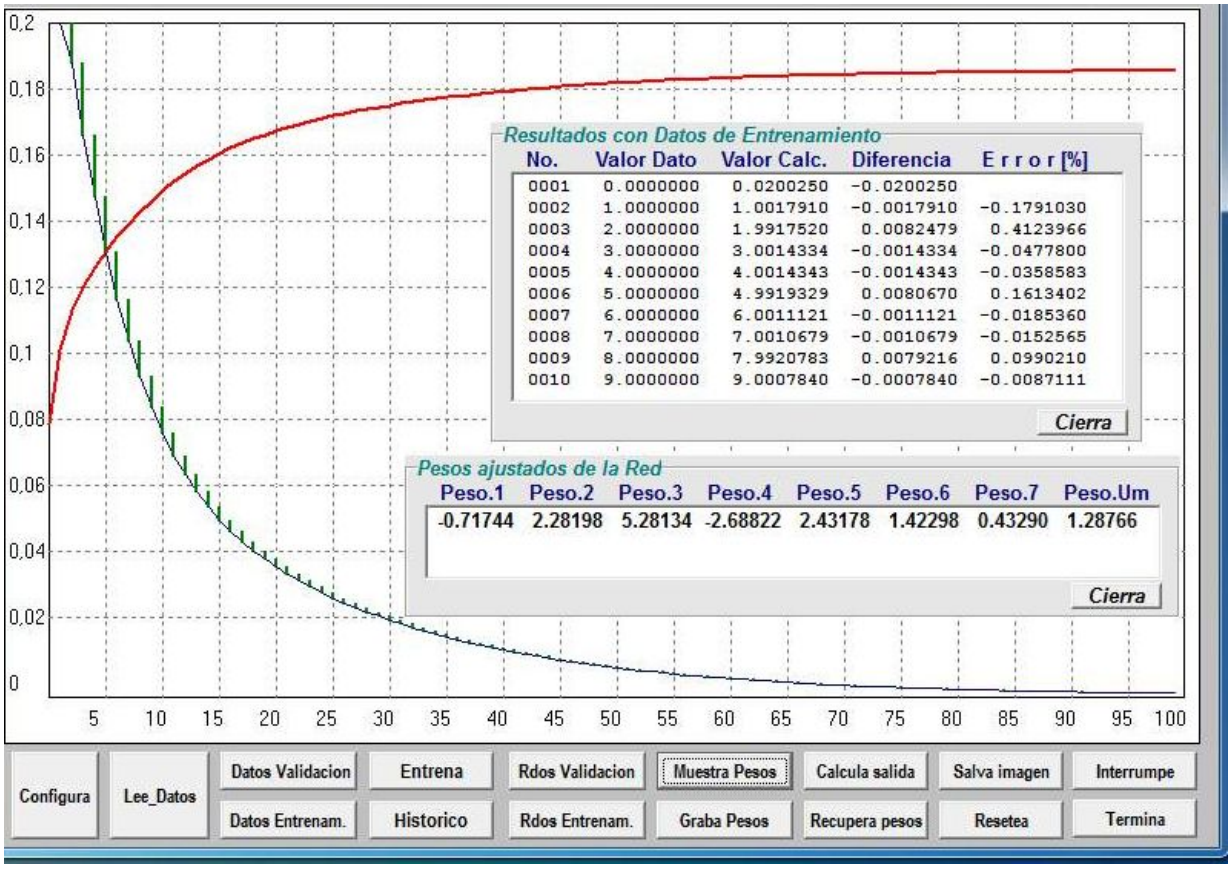
Como puede apreciarse, se trata de ocho señales (identificadas entre “0” y “7”) que toman valores de “0” o “1” según cada tecla se encuentre libre (“0”) o presionada (“1”), son afectadas por los correspondientes pesos sinápticos y procesadas por el perceptrón. La función de activación debe necesariamente ser lineal para posibilitar salidas entre “0” y “9”, es decir, valores enteros. Una vez planteado el problema, es necesario emplear el procedimiento ya estudiado en el caso de la compuerta “OR” para ajustar los pesos sinápticos, que implica que el modelo aprenda a comportarse según su fin específico. Otro aspecto a considerar es el ***factor de aprendizaje “α”***, que por esperarse una salida continua debe tener un valor apropiado que modere el proceso de ajustes de pesos. Esta unidad toma el nombre de ***adaline***.

Como puede anticiparse, dada la cantidad de condiciones entrada-salida previstas (una por cada tecla), cada ciclo de ajuste debe contemplar diez ajustes de pesos y estos ciclos deben repetirse hasta alcanzar el desempeño esperado.

Naturalmente, este proceso no es apropiado para ser realizado manualmente, por lo que es conveniente definir un algoritmo que realice los sucesivos ciclos de ajuste. Este debe operar hasta que el error medio cuadrático de un ciclo sea menor de un valor prefijado o se alcancen condiciones que demuestren la inutilidad de proseguirlo. Estas condiciones son normalmente dos: por exceder una cantidad de ciclos considerada límite o por comprobarse la estabilización del error en un valor excesivo. Estas son las habituales.

Para interpretar con facilidad el proceso de entrenamiento es muy conveniente representarlo con un gráfico. Normalmente, se expresa en la abscisa la cantidad de ajustes y en las ordenadas el error obtenido, que suele ser acompañado de la representación de un vector que representa a los pesos (raíz cuadrada de la suma de los cuadrados de los pesos sinápticos). Esta curva pone de manifiesto la convergencia del proceso hacia la solución, mostrando una evolución asintótica de la curva “pesos”. En la Figura 9 se presenta una aplicación de desarrollo del autor específica para este fin. La línea descendente muestra la evolución del error y la curva color rojo el módulo del vector “pesos”. En la primera ventana se muestran los valores de referencia y los valores calculados para las salidas. En la segunda ventana se presentan los ocho pesos ajustados: pesos sinápticos de las entradas y umbral de activación. Para el proceso de entrenamiento se hicieron cien ciclos de ajustes de pesos.

Figura 9: Representación del proceso de entrenamiento de un perceptrón y resultados



Fuente: elaboración propia.

Puede observarse que el ajuste de pesos tiene un comportamiento claramente asintótico, lo que se comprueba tanto en la evolución del error como en el módulo del vector pesos.

Se invita al lector a comprobar que los pesos obtenidos satisfacen la tabla de entradas presentada juntamente con el esquema del modelo en la figura 8. Tomando como ejemplo a las teclas “3” y “7”, puede comprobarse que a partir de sus entradas activas se llega a las siguientes expresiones:

$$y = wo + w1.x1 + w5.x5 = 1,28766 - 0,71744 + 2,43178 = 3,00200 \approx 3$$

$$y = wo + w3.x3 + w7.x7 = 1,28766 + 5,28134 + 0,43290 = 7,00190 \approx 7$$

Si el problema fue resuelto correctamente, la comprobación debería poder extenderse a cualquiera de los dígitos, obteniéndose similares resultados. Esta comprobación invita a plantear el problema en conjunto, de manera matricial. Para ello hay que reconocer una matriz principal "de activación" **[X]** con unos y ceros, que definen las señales que afectan a cada dígito. Se completa con un vector **[W]** de pesos y un vector término independiente **[V]** que contendrá los valores de las teclas. Se obtiene así:

$$[X].\{W\} = \{V\}$$

Es importante observar que el vector **{X}** tiene 10 elementos, el vector **{V}** 8 elementos y **[W]** es una matriz de 10 filas y 8 columnas. Es decir que, tal como está planteado, se trata de un sistema de diez ecuaciones y ocho incógnitas, que está sobredefinido. A partir del algebra matricial se puede hacer lo siguiente:

$$[X]^T.[X].\{W\} = [X]^T.\{V\}$$

$$\{W\} = ([X]^T.[X])^{-1}.[X]^T.\{V\}$$

$$\{W\} = [X]^{\dagger}\{V\}$$

$$\textbf{donde } [X]^{\dagger} = ([X]^{T}.[X])^{-1}.\ [X]^{T}$$

La matriz $[X]^{\dagger}$ es la psudoinversa de **[X]**, una generalización de la matriz inversa de **[X],** de tal forma que $[X]^{\dagger}\ [X] = [I]$ y $[X]^{\dagger} = [X]^{-1}$ cuando la matriz es cuadrada. A través de ese simplísimo razonamiento, se llegó a plantear la pseudoinversa de Moore (1920) y Penrose (1955), lo que permite obtener para un sistema sobredefinido una solución equivalente a la de mínimos cuadrados, una solución óptima. Se trata de una herramienta muy valiosa para la inteligencia artificial y para resolver una enorme cantidad de problemas, por lo que resulta sorprendente comprobar que todavía hoy sea difícilmente incluida en los programas de Álgebra. A pesar de haber sido planteada hace ya 100 años.

Para respaldar este alegato, se resolverá el cálculo del vector pesos que define el modelo del pad numérico por este medio. Se invita al lector a obtener $[X]^{\dagger}$ a través de alguno de los numerosos online calculators disponible en la web o, mejor aún, a que implemente el algoritmo de cálculo de la pseudoinversa de una matriz. Los resultados parciales no se presentan aquí por razones de espacio. Finalmente se obtendrá:

$$\{W\} = [X]^{\dagger}\{V\}$$

$$\{W\}^{T} = \{-1{,}1588\ 1{,}8415\ 4{,}8418\ -3{,}1590\ 1{,}7889\ 0{,}7905\ -0{,}2109\ 2{,}3685\}$$

Como puede fácilmente comprobarse, el vector {W} ahora obtenido es diferente del calculado a través del modelo neuronal. Sin embargo, tomando

el mismo ejemplo de las teclas “3” y “7”, puede también comprobarse que a partir de sus entradas activas se obtienen los valores correctos:

$$y = w_o + w_1.x_1 + w_5.x_5 = 2{,}3685 - 1{,}1588 + 1{,}7889 = 2{,}9986 \approx 3$$

$$y = w_o + w_3.x_3 + w_7.x_7 = 2{,}3685 + 4{,}8418 - 0{,}2109 = 6{,}9994 \approx 7$$

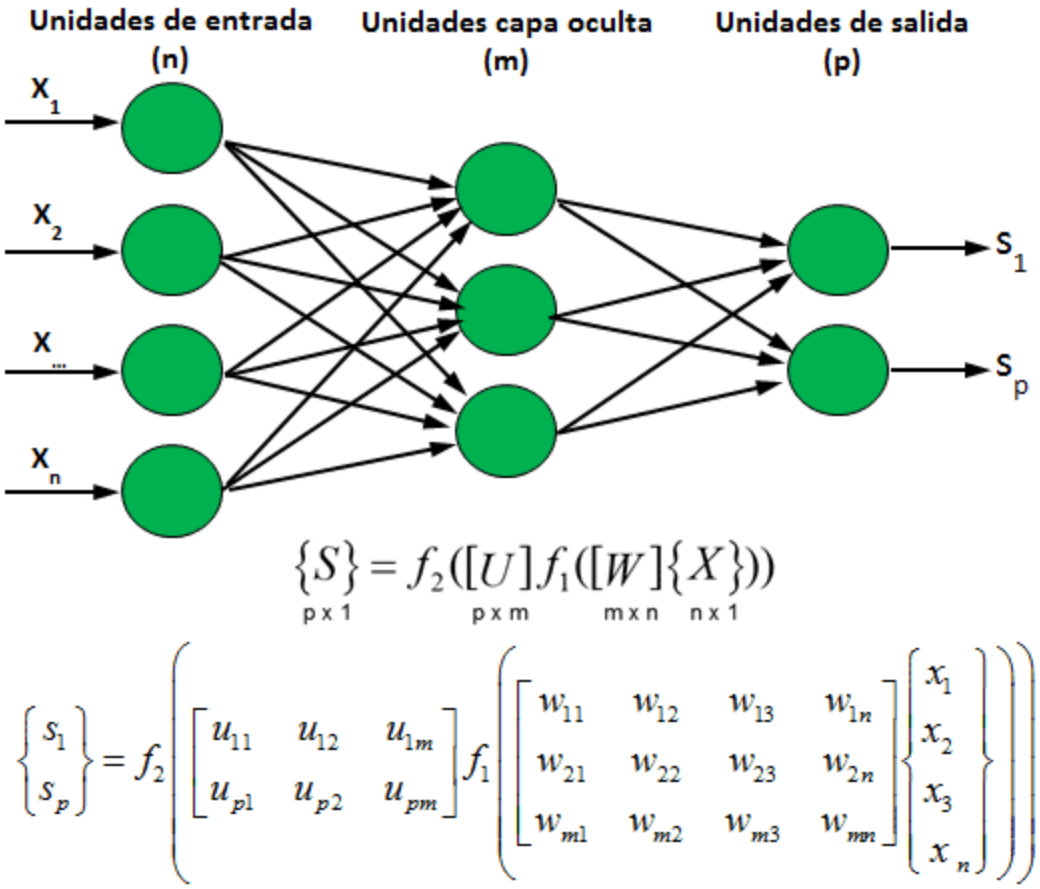
Esto significa que el problema planteado no tiene una solución única, lo que era predecible por disponerse de más ecuaciones que incógnitas. La solución obtenida a través del modelo neuronal depende de los valores de los pesos iniciales, que son aleatorios. Por lo tanto, el proceso converge a un error mínimo local que satisface las condiciones requeridas. Por el contrario, la solución algebraica satisface globalmente la condición planteada con mínimos cuadrados, por lo que es óptima.

CONTINUAR

Redes neuronales multicapa: aprendizaje supervisado

Cuando el problema conduce a sistemas de ecuaciones algebraicas que no son linealmente independientes o responde a modelos matemáticos complejos, queda fuera del alcance de un perceptrón simple. En esos casos, se utilizan varias unidades dispuestas en dos o más capas, asignando a las unidades de las capas funciones de activación continuas: sigmoideas y/o lineales. En la figura 10 se presenta un ejemplo de red multicapa de perceptrones.

Figura 10: Modelo multicapa y relaciones entre entradas, pesos y salidas



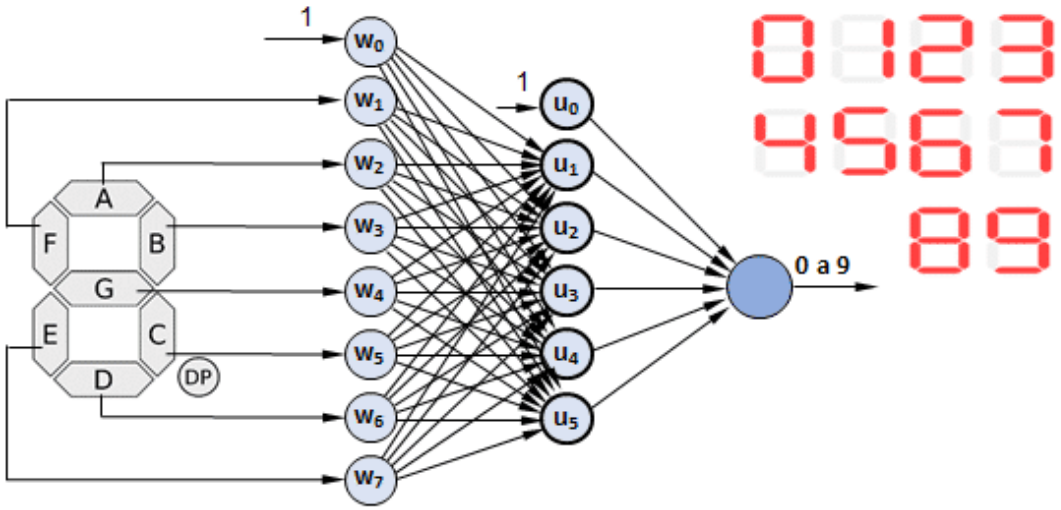
Fuente: [Imagen sin título sobre modelo multicapa]. (s.f.).

La técnica de entrenamiento de estas redes fue propuesta por Rumelhart (1986) y su nombre (*backpropagation*) proviene de la necesidad de ajustar los pesos desde las unidades de salida hacia las de entrada, de manera de distribuir las correcciones en proporción a las contribuciones al error que ha tenido cada una. La propiedad que exhiben las redes de perceptrones, con estructuras de por lo menos tres capas, de ser capaces de reproducir cualquier función genérica continua fue formalmente demostrada por

Kolmogorov y otros varios investigadores. Los detalles de esta técnica quedan fuera del alcance del curso.

Para ilustrar su uso se presenta un ejemplo muy simple, en el que el problema está representado por un sistema de ecuaciones lineales, pero las mismas no son linealmente independientes, por lo que exceden las posibilidades de un perceptrón. Se trata de la necesidad de identificar el número representado por un dígito luminoso de “siete segmentos” a partir de los elementos activados, tal como se muestra en la figura 11.

Figura 11: Esquema de la representación del dígito de siete segmentos



Entradas correspondientes a líneas de cada segmento

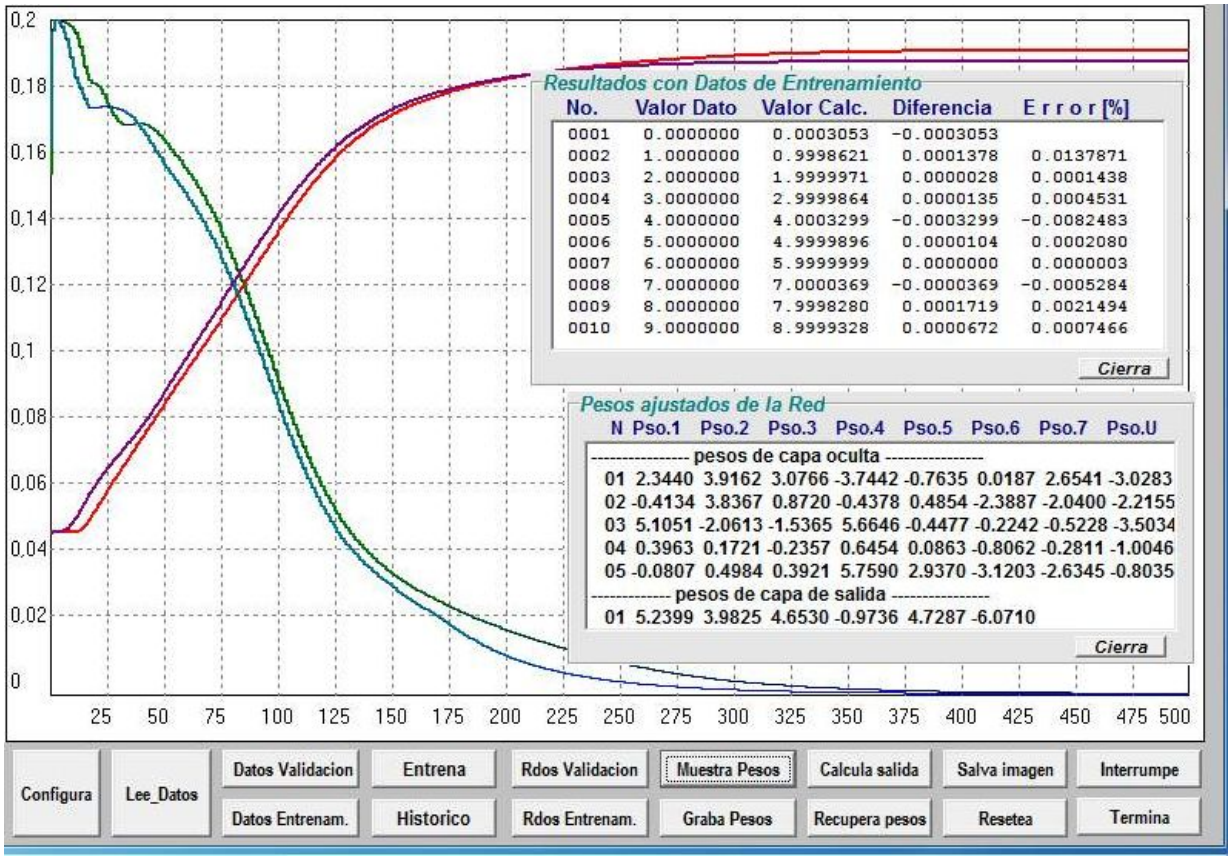
1	X_1	X_2	X_3	X_4	X_5	X_6	X_7	Salida
1	1	1	1	0	1	1	1	0
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	1	1	2
1	0	1	1	1	1	1	0	3
1	1	0	1	1	1	0	0	4
1	1	1	0	1	1	1	0	5
1	1	1	0	1	1	1	1	6
1	0	1	1	0	1	0	0	7
1	1	1	1	1	1	1	1	8
1	1	1	1	1	1	1	0	9

Fuente: [Imagen sin título sobre esquema de representación del dígito siete]. (s.f.).

El “aprendizaje” del modelo consiste en el ajuste de sus pesos, para lo cual debe definirse primero su arquitectura. Se sabe que se trata de ocho entradas, una de ellas correspondiente al escalón de activación, y una salida continua en el rango [0...9]. Luego la red tendrá siete entradas y una unidad en la capa de salida. Se opta por una capa oculta a la que se le asigna cinco unidades con función de activación sigmoidal y función lineal para la unidad de salida. Se propone un factor de aprendizaje inicial **(α)** pequeño y se ajusta

su valor conforme a los resultados obtenidos, hasta alcanzar un desempeño apropiado (ver figura 12).

Figura 12: Proceso de entrenamiento de una red de perceptrones multicapa



Fuente: elaboración propia.

Se muestran dos procesos de entrenamiento que solo difieren en los valores iniciales (aleatorios) de los pesos. Las curvas verde y azul representan el error en cada ciclo, mientras que las curvas marrón y roja representan el módulo del vector “pesos”. El comportamiento asintótico de las cuatro curvas

demuestra un proceso de entrenamiento sin dificultades. Como puede observarse en la ventana superior, el máximo error cometido es del 0,013% y corresponde a la definición del “1”. En la ventana inferior se muestran los pesos, que corresponden a las ocho entradas que impactan sobre las cinco unidades de la capa oculta y los seis pesos que impactan sobre las salidas de la capa oculta y entrada a la unidad de la capa de salida. Un total de 46 pesos cuyo ajuste demandó 500 ciclos, a partir de lo cual el ajuste en los pesos era insignificante. Una vez entrenado, el modelo posibilita la determinación del valor mostrado por el dígito a partir de los segmentos activados. Una forma no convencional de ver lo que está sucediendo, a través de un cálculo que incluye valores de entrada y pesos sinápticos.

En los casos en que el proceso de entrenamiento no tenga el desempeño exitoso aquí descripto, se deben explorar variantes en la arquitectura de la red, probando con diferente cantidad de unidades en la capa oculta. Si el problema persiste habrá que incorporar una nueva capa oculta, lo que es frecuente en el caso de la representación de funciones muy complejas. Esto obliga a ensayar con diferente cantidad de unidades para cada capa y probar otras funciones de activación. Y en cada caso hay que variar el factor de aprendizaje. Un proceso muy lento y laborioso que no reconoce recomendaciones definitivas y se apoya en la experiencia y en un trabajo muy ordenado. Debe tenerse aquí en cuenta que estos entrenamientos requieren muchos miles de ciclos de ajuste, motivo por el cual su representación gráfica es indispensable para una correcta interpretación de su desempeño.

Desafío propuesto a partir de las notas anteriores

Esta lectura se ocupó del aprendizaje supervisado en el contexto de redes neuronales artificiales hacia adelante y no recurrentes. Estas redes incluyen a las denominadas multicapas de perceptrones, adecuadas para representar funciones matemáticas altamente complejas que, sin duda alguna, son las más difundidas. Representaron la oportunidad para tomar contacto con un proceso de aprendizaje que, a partir del ajuste progresivo de una gran cantidad de parámetros, hace posible la réplica de comportamientos muy variados y complejos. Fueron los primeros modelos numéricos inspirados en los descubrimientos de Ramón y Cajal, y también las que fueron descalificadas a partir de la evaluación del desempeño de una de sus formas más simples: una unidad o perceptrón.

Cuando esta red reproduce el desempeño de sistemas de ecuaciones lineales, los procesos de entrenamiento son muy sencillos y el problema puede también ser resuelto algebraicamente a través de su matriz pseudoinversa: recurso muy efectivo y sorprendentemente poco difundido.

Por lo expuesto, estos modelos simples de entrenamiento supervisado son muy apropiados para su abordaje a través de algún algoritmo de desarrollo propio. Si bien la web tiene infinidad de pequeñas aplicaciones con este fin, se invita al lector (futuro profesional de sistemas) a desarrollar e implementar sus propias herramientas. Se presenta una oportunidad que debe ser aprovechada. El panorama cambia al ingresar al campo de las redes multicapa. Si bien sigue siendo muy recomendable que hagan sus propios desarrollos, el algoritmo *backpropagation* no es simple y no

alcanzamos a presentarlo en el curso. La inquietud queda planteada y cada uno evaluará la conveniencia de hacerlo.

Por último, se lo invita a volver sobre las preguntas aquí formuladas y las respuestas propuestas con un espíritu de autocontrol final. La intención es estimular una mirada crítica al trabajo realizado desde una posición de mayores conocimientos sobre los temas tratados.

Se reitera que estas preguntas tienen como única finalidad contribuir a que el estudiante compruebe por sí mismo sus conocimientos, no constituyen una instancia de evaluación.

Las consignas son las siguientes:

1. Revise la clasificación de los modelos neuronales para luego identificar y justificar la forma en que queda encuadrado el perceptrón y *adaline*.
2. Haga lo mismo con otros modelos conocidos como “de aprendizaje supervisado”.
3. Identifique los componentes básicos de una unidad neuronal artificial y los elementos de la neurona biológica.
4. Identifique los problemas que pueden ser resueltos por una unidad neuronal simple y los que quedan fuera de su alcance. Justifique.

5. Identifique los problemas que están al alcance de ser resueltos por la pseudoinversa de la matriz principal del sistema. Justifique.

CONTINUAR

Referencias

Hernández Paxtián, Z.J. (2011). Biología + electrónica: ¿es posible replicarnos? Recuperado de https://www.rics.org.mx/index.php/RICS/article/view/12/97#:~:text=La%20uni%C3%B3n%20entre%20dos%20neuronas%20se%20denomina%20sinapsis.&text=En%20estado%20de%20reposo%20el,2001%3B%20Stratton%2C%201984).

Red neuronal. (s.f.). Definición de red neuronal. Recuperado de https://sites.google.com/site/inteligenciascarol/red-neuronal?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1#:~:text=Un%20ax%C3%B3n%20se%20define%20como,los%20axones%20de%20otras%20neuronas.

CONTINUAR

Descarga en PDF

PDF

Módulo 3 - Lectura 3.pdf

933.9 KB