



# INSTITUTO POLITÉCNICO NACIONAL CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

No. 125

Serie: VERDE

Fecha: 2006

## Algoritmo Backpropagation para Redes Neuronales: conceptos y aplicaciones

Marco Antonio Valencia Reyes<sup>1</sup>

Cornelio Yáñez Márquez<sup>2</sup>

Luis Pastor Sánchez Fernández<sup>3</sup>

### RESUMEN

En este trabajo se expone un breve panorama de la historia de las redes neuronales, siendo el tema principal de este trabajo las redes neuronales backpropagation, por lo que se explica el funcionamiento de este tipo de red y se da un ejemplo empleando un software llamado NeuroSolutions.

Esta publicación es un producto de los proyectos de investigación diseñados y desarrollados en el Centro de Investigación en Computación del IPN.

**Palabras clave:** Redes Neuronales, Back Propagation, NeuroSolutions.

**Agradecimientos:** Los autores agradecen el apoyo que recibieron de las siguientes instituciones, para la realización de este trabajo: SIP, COFAA y Secretaría Académica del Instituto Politécnico Nacional, CONACyT y Sistema Nacional de Investigadores.

<sup>1</sup>Estudiante de la Maestría en Ingeniería de Cómputo del CIC-IPN mvalenciab05@sagitario.cic.ipn.mx

<sup>2</sup>Profesor-Investigador del CIC-IPN cyanez@cic.ipn.mx

<sup>3</sup>Profesor-Investigador del CIC-IPN lsanchez@cic.ipn.mx

## ÍNDICE

	Pág.
1 Introducción	1
2 Historia de las Redes Neuronales	1
3 El Modelo Biológico	3
4 Elementos de una Red Neuronal Artificial	3
5 La Red Backpropagation	5
6 Estructura y Aprendizaje de la Red Backpropagation	6
7 Pasos para aplicar el Algoritmo de Entrenamiento	6
8 Número de Capas Ocultas	8
9 Caso de Estudio	8
10 Bibliografía	12

## Introducción

El hombre se ha caracterizado por la búsqueda constante de nuevas vías para mejorar sus condiciones de vida, esto le ha servido para reducir el trabajo en aquellas operaciones en las que la fuerza juega un papel primordial. Los progresos obtenidos han permitido dirigir sus esfuerzos a otros campos, como por ejemplo, a la construcción de máquinas calculadoras que ayuden a resolver de forma automática y rápida determinadas operaciones que resultan tediosas cuando se realizan a mano.

En 1946 fue construida la primera computadora electrónica, ENIAC. Desde entonces los desarrollos en este campo han tenido un auge espectacular. Las computadoras permiten implementar fácilmente algoritmos para resolver multitud de problemas que antes resultaban engorrosos de resolver. Sin embargo, se observa una limitación importante: ¿qué ocurre cuando el problema que se quiere resolver no admite un tratamiento algorítmico, como es el caso, de la clasificación de objetos por rasgos comunes? Este ejemplo demuestra que la construcción de nuevas máquinas versátiles requiere un enfoque del problema desde otro punto de vista. Los desarrollos actuales de los científicos se dirigen al estudio de las capacidades humanas como una fuente de nuevas ideas para el diseño de nuevas y mejores máquinas[1].

Las redes neuronales son más que otra forma de emular ciertas características propias de los humanos, como la capacidad de memorizar y de asociar hechos. Si se examinan con atención aquellos problemas que no pueden expresarse a través de un algoritmo, se observará que todos ellos tienen una característica en común: la experiencia. El hombre es capaz de resolver estas situaciones acudiendo a la experiencia acumulada. Así, parece claro que una forma de aproximarse al problema consista en la construcción de sistemas que sean capaces de reproducir esta característica humana.

Las redes neuronales son un modelo artificial y simplificado del cerebro humano, que es el ejemplo perfecto del que se dispone para un sistema que es capaz de adquirir conocimiento a través de la experiencia. Una red neuronal es “un nuevo sistema para el tratamiento de la información, cuya unidad básica de procesamiento está inspirada en la célula fundamental del sistema nervioso humano: *la neurona*”. Todos los procesos del cuerpo humano se relacionan en alguna u otra forma con la (in)actividad de estas neuronas. Las mismas son un componente relativamente simple del ser humano, pero cuando millares de ellas se conectan en forma conjunta se hacen muy poderosas.

## Historia de las Redes Neuronales

**1943 – Warren S. McCulloch/Walter Pitts.** Un neurofisiólogo y un matemático, respectivamente, fueron los primeros teóricos que concibieron los fundamentos de las redes neuronales, al publicar una teoría acerca de la forma de trabajar de las neuronas (Un Cálculo Lógico de la Inminente Idea de la Actividad Nerviosa - Boletín de Matemática Biofísica 5: 115-133). Ellos realizaron el primer modelo de neurona artificial.

**1949 - Donald O. Hebb.** Fue el primero en explicar los procesos del aprendizaje (que es el elemento básico de la inteligencia humana) desde un punto de vista psicológico, desarrollando una regla de como el aprendizaje ocurría. Aun hoy, este es el fundamento de la mayoría de las funciones de aprendizaje que pueden hallarse en una red neuronal. Su idea fue que el aprendizaje ocurría cuando ciertos cambios en una neurona eran activados.

**1957 - Frank Rosenblatt.** Comenzó el desarrollo del Perceptrón. Esta es la red neuronal más antigua; utilizándose hoy en día para aplicación como reconocedor de patrones. Este modelo era capaz de generalizar, es decir, después de haber aprendido una serie de patrones podía reconocer otros similares, aunque no se le hubiesen presentado anteriormente. Sin embargo, tenía una serie de limitaciones, por ejemplo, su incapacidad para resolver el problema de la función OR-exclusiva y, en general, era incapaz de clasificar clases no separables linealmente.

**1960 - Bernard Widrow/Marcial Hoff.** Desarrollaron el modelo Adaline (ADAdaptive LINear Elements). Esta fue la primera red neuronal aplicada a un problema real (filtros adaptativos para eliminar ecos en las líneas telefónicas) que se ha utilizado comercialmente durante varias décadas.

**1961 - Karl Steinbuch:** Desarrolla la Lernmatrix (memoria asociativa que es capaz de recuperar una imagen o conocimiento, aún cuando el medio este viciado).

**1969 - Marvin Minsky/Seymour Papert.** En este año surgieron críticas que frenaron, hasta 1982, el crecimiento que estaban experimentando las investigaciones sobre redes neuronales. Minsky y Papert, del Instituto Tecnológico de Massachussets (MIT), publicaron un libro *Perceptrons*, en el cual probaron (matemáticamente) que el Perceptrón no era capaz de resolver problemas relativamente fáciles, tales como el aprendizaje de una función no-lineal. A pesar del libro, algunos investigadores continuaron su trabajo. Tal fue el caso de **James Anderson**, que desarrolló un modelo lineal, llamado **Lineal Associator**.

**1974 - Paul Werbos.** Desarrolló la idea básica del algoritmo de aprendizaje de *propagación hacia atrás* (backpropagation); cuyo significado quedó definitivamente aclarado en 1985.

**1977 - Teuvo Kohonen.** Ingeniero electrónico de la Universidad de Helsinki, desarrolló un modelo similar al de Anderson, pero independientemente.

**1982 - John J. Hopfield.** Provocó el renacimiento de las redes neuronales con su libro: "Computación neuronal de decisiones en problemas de optimización."

**1986 - Rumelhart/Hinton.** Redescubrieron el algoritmo de aprendizaje de propagación hacia atrás (backpropagation). A partir de 1986, el panorama fue alentador con respecto a las investigaciones y el desarrollo de las redes neuronales[1].

## El Modelo Biológico

Las neuronas y las conexiones entre ellas (sinápsis) constituyen la clave para el procesamiento de la información. De alguna manera la neurona elabora una señal de salida a partir de las dendritas, que son la vía de entrada de las señales que se combinan en el cuerpo de la neurona, las envía al axón, que es el camino de salida de la señal generada por la neurona[1].

Las sinapsis, son las unidades funcionales y estructurales elementales que median entre las interacciones de las neuronas. En las terminaciones de las sinapsis se encuentran unas vesículas que contienen unas sustancias químicas llamadas neurotransmisores, que ayudan a la propagación de las señales electroquímicas de una neurona a otra[2].

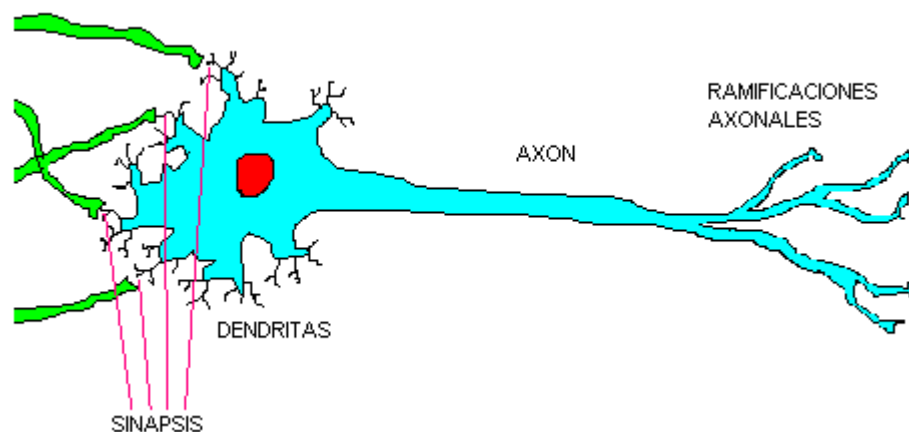


Figura 1. Neurona y Conexiones sinápticas

Lo que básicamente ocurre en una neurona biológica es lo siguiente: la neurona es estimulada o excitada a través de sus entradas y cuando se alcanza un cierto umbral, la neurona se dispara o activa, pasando una señal hacia el *axón*.

Como ya se sabe, el pensamiento tiene lugar en el cerebro, que consta de billones de neuronas interconectadas. Así, el secreto de la “inteligencia” -sin importar como se defina- se sitúa dentro de estas neuronas interconectadas y de su interacción. La forma que dos neuronas interactúan no está totalmente conocida. En general, una neurona envía su salida a otras por su axón, el axón lleva la información por medio de diferencias de potencial, u ondas de corriente, que dependen del potencial de la neurona. La neurona recoge las señales por su sinápsis sumando todas las influencias excitadoras e inhibidoras. Si las influencias excitadoras positivas dominan, entonces la neurona da una señal positiva y manda este mensaje a otras neuronas por sus sinápsis de salida.

## Elementos de una Red Neuronal Artificial

Las redes neuronales son modelos que intentan reproducir el comportamiento del cerebro[3]. Los mismos constan de dispositivos elementales de proceso: *las neuronas*. A partir de ellas, se

pueden generar representaciones específicas, de tal forma que un estado conjunto de ellas puede significar una letra, un número u otro objeto. Generalmente se pueden encontrar tres tipos de neuronas:

Aquellas que reciben estímulos externos relacionados con el aparato sensorial, que tomarán la información de entrada. Dicha información se transmite a ciertos elementos internos que se ocupan de su procesamiento.

Es en las *sinapsis* y *neuronas* correspondientes a este segundo nivel donde se genera cualquier tipo de representación interna de información. Como no tienen relación directa con la información de entrada ni con la salida, estos elementos se denominan *unidades ocultas*.

Una vez finalizado el período de procesamiento, la información llega a las unidades de salida, cuya misión es dar la respuesta al sistema.

A continuación se puede ver el esquema de una red neuronal[1][3]:

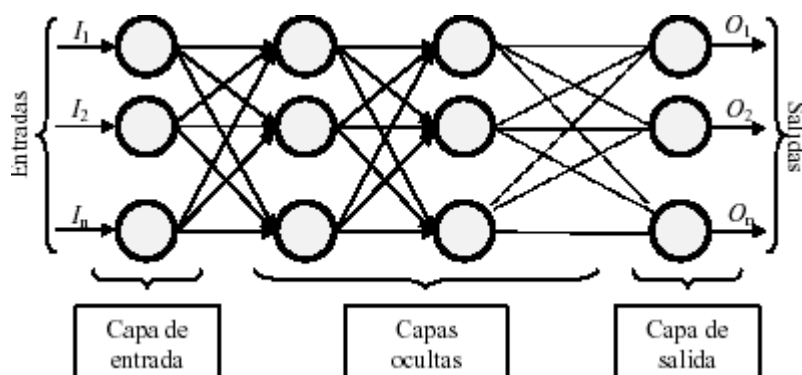


Figura 2. Esquema de una Red Neuronal

El esquema está constituido por neuronas interconectadas y arregladas en tres capas (esto último puede variar). Los datos ingresan por medio de la “capa de entrada”, pasan a través de la “capa oculta” y salen por la “capa de salida”. Cabe mencionar que la capa oculta puede estar constituida por varias capas a su vez.

En la siguiente figura se compara una neurona biológica con una neurona artificial. En la misma se pueden observar las similitudes entre ambas (tienen entradas, utilizan pesos y generan salidas).

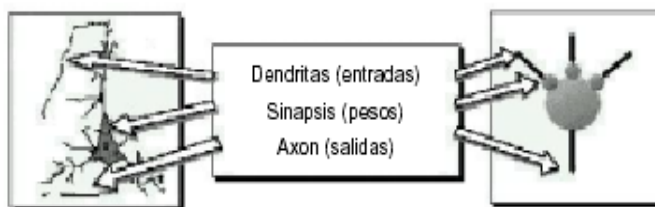
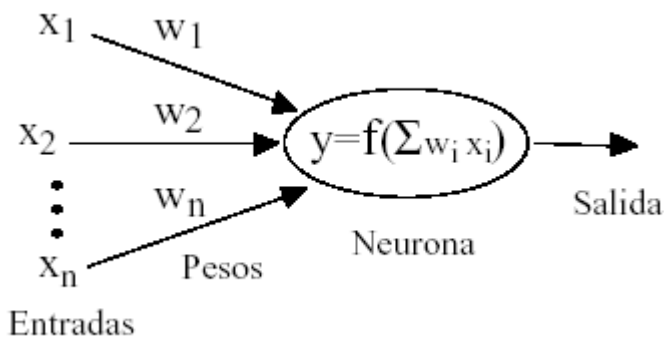


Figura 3. Similitudes entre una Neurona Biológica y una Artificial

La neurona artificial pretende imitar las características más importantes de las neuronas biológicas. Cada neurona  $i$ -ésima está caracterizada en cualquier instante por un valor numérico denominado valor o estado de activación  $a_i(t)$ ; asociado a cada unidad, existe una función de salida,  $f_i$ , que transforma el estado actual de activación en una señal de salida. Dicha señal es enviada a través de los canales de comunicación unidireccionales a otras unidades de la red; en estos canales la señal se modifica de acuerdo con la sinápsis (el peso,  $w_{ji}$ ) asociada a cada uno de ellos según determinada regla. Las señales moduladas que han llegado a la unidad  $j$ -ésima se combinan entre ellas, generando así la entrada total  $Net_j$ .

$$Net_j = \sum_i y_i w_{ji}$$

Una función de activación,  $F$ , determina el nuevo estado de activación  $a_j(t+1)$  de la neurona, teniendo en cuenta la entrada total calculada y el anterior estado de activación  $a_j(t)$ .



### La Red Backpropagation

En 1986, Rumelhart, Hinton y Williams, formalizaron un método para que una red neuronal *aprendiera* la asociación que existe entre los patrones de entrada y las clases correspondientes, utilizando varios niveles de neuronas.

El método backpropagation (propagación del error hacia atrás), basado en la generalización de la regla delta, a pesar de sus limitaciones, ha ampliado de forma considerable el rango de aplicaciones de las redes neuronales. El funcionamiento de la red backpropagation (BPN) consiste en el aprendizaje de un conjunto predefinido de pares de entradas-salidas dados como ejemplo: primero se aplica un patrón de entrada como estímulo para la primera capa de las neuronas de la red, se va propagando a través de todas las capas superiores hasta generar una salida, se compara el resultado en las neuronas de salida con la salida que se desea obtener y se calcula un valor de error para cada neurona de salida. A continuación, estos errores se transmiten hacia atrás, partiendo de la capa de salida hacia todas las neuronas de la capa intermedia que contribuyan directamente a la salida. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido un error que describa su aportación relativa al error total. Basándose en el valor del error recibido, se reajustan los pesos de conexión de cada neurona, de

manera que en la siguiente vez que se presente el mismo patrón, la salida esté más cercana a la deseada[1][3][4].

La importancia de la red backpropagation consiste en su capacidad de autoadaptar los pesos de las neuronas de las capas intermedias para *aprender* la relación que existe ente un conjunto de patrones de entrada y sus salidas correspondientes. Es importante la capacidad de *generalización*, facilidad de dar salidas satisfactorias a entradas que el sistema no ha visto nunca en su fase de entrenamiento. La red debe encontrar una representación interna que le permita generar las salidas deseadas cuando se le dan entradas de entrenamiento, y que pueda aplicar, además, a entradas no presentadas durante la etapa de aprendizaje para clasificarlas.

### Estructura y Aprendizaje de la Red Backpropagation

En una red Backpropagation existe una capa de entrada con  $n$  neuronas y una capa de salida con  $m$  neuronas y al menos una capa oculta de neuronas internas. Cada neurona de una capa (excepto las de entrada) recibe entradas de todas las neuronas de la capa anterior y envía su salida a todas las neuronas de la capa posterior (excepto las de salida). No hay conexiones hacia atrás *feedback* ni laterales entre las neuronas de la misma capa.

La aplicación del algoritmo tiene dos fases, una hacia delante y otra hacia atrás. Durante la primera fase el patrón de entrada es presentado a la red y propagado a través de las capas hasta llegar a la capa de salida. Obtenidos los valores de salida de la red, se inicia la segunda fase, comparándose éstos valores con la salida esperada para así obtener el error. Se ajustan los pesos de la última capa proporcionalmente al error. Se pasa a la capa anterior con una retropropagación del error, ajustando los pesos y continuando con este proceso hasta llegar a la primera capa. De esta manera se han modificado los pesos de las conexiones de la red para cada patrón de aprendizaje del problema, del que conocíamos su valor de entrada y la salida deseada que debería generar la red ante dicho patrón.

La técnica *Backpropagation* requiere el uso de neuronas cuya función de activación sea continua, y por lo tanto, diferenciable. Generalmente, la función utilizada será del tipo sigmoidal.

### Pasos para aplicar el Algoritmo de Entrenamiento

Paso 1. Inicializar los pesos de la red con valores pequeños aleatorios.

Paso 2. Presentar un patrón de entrada y especificar la salida deseada que debe generar la red.

Paso 3. Calcular la salida actual de la red. Para ello se presentan las entradas a la red y se calcula la salida de cada capa hasta llegar a la capa de salida, ésta será la salida de la red. Los pasos son los siguientes:

Se calculan las entradas netas para las neuronas ocultas procedentes de las neuronas de entrada.

Para una neurona  $j$  oculta:

$$net_{pj}^h = \sum_{i=1}^N w_{ji}^h x_{pi} + \theta_j^h$$



en donde el índice  $h$  se refiere a magnitudes de la capa oculta; el subíndice  $p$ , al  $p$ -ésimo vector de entrenamiento, y  $j$  a la  $j$ -ésima neurona oculta. El término  $\theta$  puede ser opcional, pues actúa como una entrada más.

Se calculan las salidas de las neuronas ocultas:  $y_{pj} = f_j^h(net_{pj}^h)$ .

Se realizan los mismos cálculos para obtener las salidas de las neuronas de salida:

$$net_{pk}^o = \sum_{j=1}^L w_{kj}^o y_{pj} + \theta_k^o$$

$$y_{pk} = f_k^o(net_{pk}^o)$$

Paso 4 . Calcular los términos de error para todas las neuronas.

Si la neurona  $k$  es una neurona de la capa de salida, el valor de la delta es:

$$\delta_{pk}^o = (d_{pk} - y_{pk}) f_k^{o'}(net_{pk}^o)$$

La función  $f$  debe ser derivable. En general disponemos de dos formas de función de salida:

La función lineal:  $f_k(net_{jk}) = net_{jk}$

La función sigmoideal:  $f_k(net_{jk}) = \frac{1}{1 + e^{-net_{jk}}}$

La selección de la función depende de la forma que se decida representar la salida: si se desea que las neuronas de salida sean binarias, se utiliza la función sigmoideal, en otros casos, la lineal.

Para una función lineal, tenemos:  $f_k^{o'} = 1$ , mientras que la derivada de una función sigmoideal es:

$f_k^{o'} = f_k^o(1 - f_k^o) = y_{pk}(1 - y_{pk})$  por lo que los términos de error para las neuronas de salida quedan:

$\delta_{pk}^o = (d_{pk} - y_{pk})$  para la salida lineal.

$\delta_{pk}^o = (d_{pk} - y_{pk}) y_{pk}(1 - y_{pk})$  para la salida sigmoideal.

Si la neurona  $j$  no es de salida, entonces la derivada parcial del error no puede ser evaluada directamente, por tanto se obtiene el desarrollo a partir de valores que son conocidos y otros que pueden ser evaluados.

La expresión obtenida en este caso es:  $\delta_{pj}^h = f_j^{h'}(net_{pj}^h) \sum_k \delta_{pk}^o w_{kj}^o$  donde observamos que el

error en las capas ocultas depende de todos los términos de error de la capa de salida. De aquí surge el término *propagación hacia atrás*.

Paso 5. Actualización de los pesos: para ello utilizamos un algoritmo recursivo, comenzando por las neuronas de salida y trabajando hacia atrás hasta llegar a la capa de entrada, ajustando los pesos de la siguiente forma:

Para los pesos de las neuronas de la capa de salida:

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \Delta w_{kj}^o(t+1)$$

$$\Delta w_{kj}^o(t+1) = \alpha \delta_{pk}^o y_{pj}$$

Para los pesos de las neuronas de la capa oculta:

$$w_{ji}^h(t+1) = w_{ji}^h(t) + \Delta w_{ji}^h(t+1)$$

$$\Delta w_{ji}^h(t+1) = \alpha \delta_{pj}^h x_{pi}$$

En ambos casos, para acelerar el proceso de aprendizaje se puede añadir un término *momento*.

Paso 6. El proceso se repite hasta que el término de error  $E_p = \frac{1}{2} \sum_{k=1}^M \delta_{pk}^2$  resulta aceptablemente pequeño para cada uno de los patrones aprendidos[4][5].

## Número de Capas Ocultas

No se pueden dar reglas concretas para determinar el número de neuronas o número de capas de una red para resolver un problema concreto.

Respecto al número de capas de la red, en general tres capas son suficientes (entrada -oculta-salida). Sin embargo, hay veces que un problema es fácil de resolver con más de una capa oculta. El tamaño de las capas, tanto de entrada como de salida, suele estar determinado por la naturaleza de la aplicación. En cambio, decidir cuántas neuronas debe tener una capa oculta no suele ser tan evidente. El número de neuronas ocultas interviene en la eficiencia de aprendizaje y de generalización de la red. No hay ninguna regla que indique el número óptimo, en cada problema se debe ensayar.

## Caso de Estudio

### Objetivo

El tipo de problema que se pretende resolver con la red construida en este ejemplo es de clasificación, siendo el objetivo de este ejemplo el entrenar una red neuronal para determinar el sexo de un espécimen de cangrejo de piedra (varón o hembra) con base en los datos suministrados en un archivo ASCII.

### Descripción de los datos

Los datos de muestreo que se usarán contienen diversos atributos de especímenes de cangrejo de piedra. Hay 50 especímenes varón y 50 especímenes hembras para cada una de las dos especies (forma azul y forma anaranjada) dando un total de 200 especímenes. Los datos fueron suministrados en un archivo ASCII que contiene el siguiente formato:

SpecimenNumber,Species,FrontalLip,RearWidth,Length,Width,Depth,Sex

142,0,20.6,14.4,42.8,46.5,19.6,Male

19,1,13.3,11.1,27.8,32.3,11.3,Male

169,0,16.7,14.3,32.3,37,14.7,Female

56,1,9.8,8.9,20.4,23.9,8.8,Female

Dónde cada una de las columnas representa la siguiente información:

1. SpecimenNumber: Numero de Especimen,
2. Species: Especie,
3. FrontalLip: Labio Frontal,
4. RearWidth: Anchura Trasera,
5. Length: Longitud,
6. Width: Anchura,
7. Depth: Profundidad, y
8. Sex: Sexo.

## **Modelado**

El software empleado para presentar el caso de estudio se llama NeuroSolutions. En NeuroSolutions los iconos circulares anaranjados se llaman axons, las conexiones entre las axons son los iconos con líneas horizontales y diagonales y se llaman synapses. NeuroSolutions usa la retropropagación del error para entrenar la red neuronal. Los iconos menores en lo alto de los axons y synapses se llaman componentes backpropagation y propagan el error hacia atrás desde el fin de la red al comienzo de la misma. Los axons de color verde en lo alto de los componentes backpropagation se llaman componentes de búsqueda del gradiente y se encargan de ajustar los pesos contenidos en las synapses y los axons, de este modo es como la red se entrena.

## **Construcción del modelo**

El modelo consiste en un conjunto de datos (tabla con los datos de entrenamiento y prueba), una herramienta de análisis y un conjunto de parámetros.

Para crear el modelo en NeuroSolutions, se debe especificar:

- 1) La base de datos que contiene los patrones de entrada a ser “aprendidos” por la red.
- 2) Se debe indicar cuáles son las entradas a la red y cuál es la salida.
- 3) El porcentaje de datos usados para entrenar.
- 4) El porcentaje de datos a ser usados después de que la red haya sido entrenada.

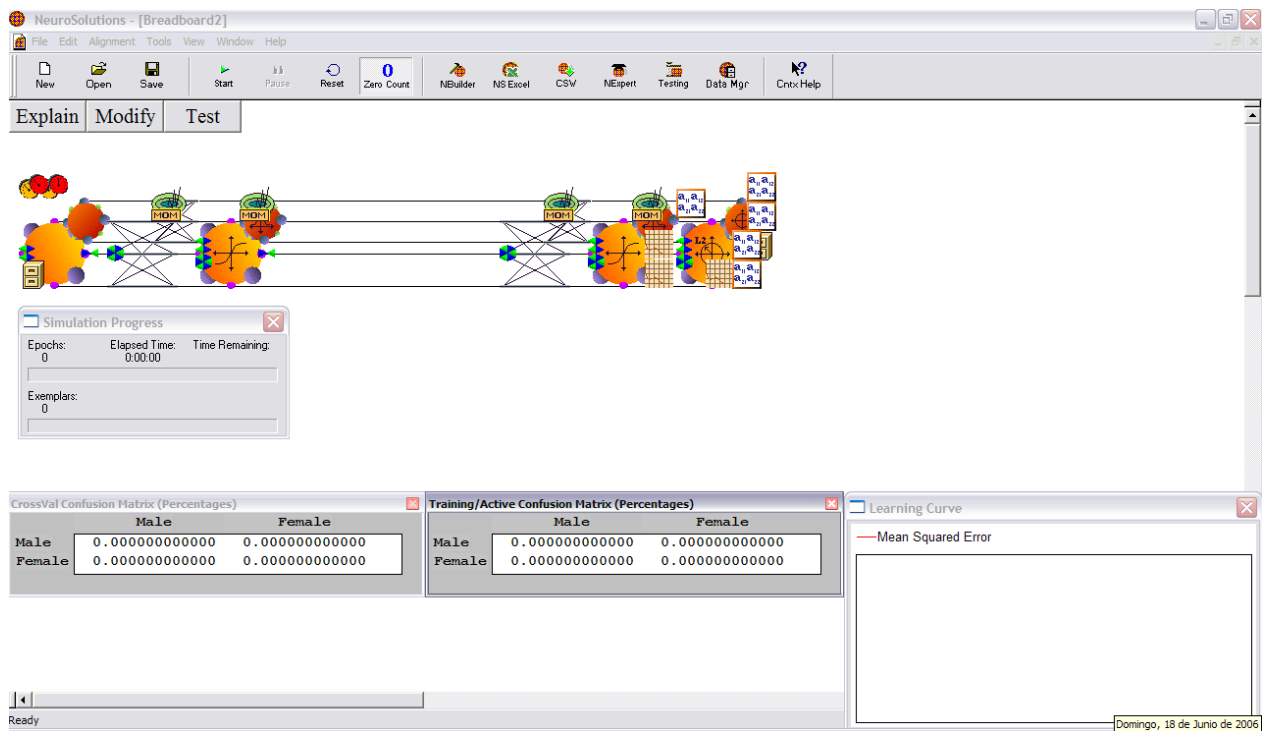


Figura 4. Modelo de Red Neuronal Backpropagation realizado con el software “NeuroSolutions”

## Ejecución del Modelo

Des Male	Des Female	Out Male	Out Female
1	0	1,027715539	-0,025608678
1	0	1,042436421	-0,042517249
0	1	0,563072098	0,454152583
1	0	0,723810849	0,27644096
0	1	-0,002725647	0,987922206
0	1	-0,049338357	1,045169246
1	0	0,645250245	0,333899846
1	0	1,043509727	-0,043673595
1	0	1,025918669	-0,009870963
0	1	0,188675701	0,797158575
1	0	0,905464009	0,120555525
1	0	1,047222752	-0,043095965
0	1	-0,049145316	1,044949895
1	0	0,998924901	0,018659744
0	1	-0,052595227	1,05010776
1	0	1,042632838	-0,042761023
1	0	1,048602499	-0,044301413
1	0	1,05391416	-0,052515702
1	0	1,047407404	-0,045685646
0	1	-0,055260165	1,054829492
0	1	0,328575752	0,656319982
0	1	-0,021645556	1,010273431
1	0	1,054177514	-0,05322226
1	0	1,052176648	-0,050905165
0	1	-0,05305996	1,050791445
0	1	0,527859428	0,509633415
0	1	0,085451848	0,899299918
1	0	1,026418464	-0,021602023
0	1	-0,025754371	1,016053794
1	0	0,420197837	0,587838167
0	1	-0,040786205	1,033850011
1	0	1,028771201	-0,025821596
1	0	1,041606183	-0,04126673
1	0	1,032510668	-0,031077732
0	1	0,099841444	0,903630727
1	0	0,727320167	0,312420909
0	1	0,059473619	0,922807144
1	0	0,349390885	0,612543341
0	1	-0,055190217	1,054667647
0	1	-0,053026496	1,050886039

Tabla 1. Datos obtenidos mediante el software “NeuroSolutions” después de haber entrenado y empleado el modelo de Red Neuronal Backpropagation diseñado

## **Bibliografía**

- [1] José R. Hilera y Victor J. Martinez, “Redes Neuronales Artificiales”, Alfaomega-Rama, 2000
- [2] S.Y. Kung, "Digital Neural Networks", Prentice Hall, 1993.
- [3] E. Castillo Ron, Á. Cobo Ortega, J. M. Gutiérrez Llorente, R. E. Pruneda González, “Introducción a la Redes Funcionales con Aplicaciones”, Paraninfo, 1999
- [4] J.A. Freeman y D.M. Skapura, "Redes Neuronales: Algoritmos, Aplicaciones y Técnicas de Programación", Addison-Wesley, 1993.
- [5] Bonifacio Martín y Alfredo Sanz, “Redes Neuronales y Sistemas Difusos”, Alfaomega-Rama, 2002