

## RESUMEN ANALITICO

### HIBRIDO ENTRE UNA RED NEURONAL Y UN MODELO ARIMA (\*)

El presente artículo es el resultado de un año de investigación con el objeto de satisfacer la necesidad de Fasecolda, por presentar los precios comerciales de los vehículos, necesarios para que las compañías de Seguros en Colombia ajusten sus primas sobre los precios comerciales de los vehículos, que satisfagan al cliente y que generen un equilibrio económico en caso de siniestro.

Por las características del problema se diseñó una Red Neuronal, como modelo que solucionara la dificultad que se estaba presentando, educar su cerebro y comenzar a arrojar resultados, fue el éxito con el que se logró satisfacer esta necesidad, sin embargo se presentaron casos, en los que no había información para alimentar la Red y hubo que acudir a aplicar un modelo ARIMA sobre la información resultante y/o utilizar un índice de suavización para la estimación del precio de un modelo de un vehículo que existe en el mercado, pero no en las neuronas de alimentación de la Red, en estas condiciones se creó el Híbrido .

La Red neuronal consiste de cinco neuronas que se educan permanentemente (bases de datos de Colserauto (\*), más de 10.000 clientes mensuales a nivel nacional), a estas se les asignó un peso igual inicialmente, utilizando las bases de datos de Colserauto, con la información pertinente origen del precio cliente y fuente, se encuentra la moda y/o mediana o precio más frecuente o deseado e igualmente se conoce el peso que tiene cada neurona, se aplica la función de transferencia obteniéndose la señal con información significativa y así mismo la diferencia observada entre el precio deseado y esta señal. Para el cálculo del peso sináptico, se multiplica esta diferencia, por la tasa de crecimiento geométrico esperado y el peso de activación inicial. El factor de suavización esperado se calcula sobre los pesos sinápticos. En caso de no existir un modelo de un vehículo se aplica este último factor y se suaviza la serie si no existe suficiente información en caso contrario se aplica un modelo ARIMA para su estimación. Si se trata de un vehículo importado de alta gama, se aplica un umbral ( $\emptyset$ ) sobre la función de salida:  $Y_t = S(t) + S(t) * \emptyset$  y se guardan las mismas condiciones, este umbral consiste en el cambio del dólar de un mes a otro.

\* Híbrido entre una Red neuronal y un modelo ARIMA: Investigación financiada por Colserauto y Fasecolda, realizada entre Ene/2004 – Feb/2005

\*\* Autor: Gerardo Ardila Duarte, Lc. Matemática, UPN. Esp. Análisis de datos USALLE. Esp. Docencia Universitaria U San Buenaventura, Estudios completos Mg. Estadística UNAL. Docente investigador FUKL, Docente UMNG, Docente ULIBRE, Estadístico Seguirs del Estado. Curso de Inteligencia Artificial convenio U Villas Cuba – FUKL.

## PALABRAS CLAVE

Neurona Artificial (RNA): Es un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Consiste en simular las propiedades observadas en los sistemas neuronales biológicos a través de modelos matemáticos creados mediante mecanismos artificiales

Funcionamiento de la red neuronal: se compone de unidades llamadas neuronas. Cada neurona recibe una serie de entradas a través de interconexiones y emite una salida. Esta salida viene dada por una función de propagación o excitación, que por lo general consiste en la sumatoria de cada entrada multiplicada por el peso de su interconexión. Si el peso es positivo, la conexión se denomina excitatoria; si es negativo, inhibitoria.

Híbrido: Se llama híbrido al animal o al vegetal procreado por dos individuos de distinta especie o raza. En este caso la Redes y los Modelo ARIMA, buscan un objetivo de estimación pero en diferentes vías.

Función de Transferencia: Es un modelo que facilita el estudio de una neurona. Las entradas a la red se presentan con un vector, que para el caso de una sola neurona contiene solo un elemento,  $w$  sigue representando los pesos y la salida neta del sumador  $n$  de la red; la salida total está determinada por la función de transferencia, la cual puede ser una función lineal o no lineal de  $n$ , y que es escogida dependiendo de las especificaciones del problema que la neurona tenga que resolver; aunque las RNA se inspiren en modelos biológicos no existe ninguna limitación para realizar modificaciones en las funciones de salida, así que se encontrarán modelos artificiales que nada tienen que ver con las características del sistema biológico.

Señal con información significativa: La relación señal/ruido (en inglés *Signal to noise ratio* SNR o S/N) se define como el margen que hay entre el nivel de referencia (información significativa) y el ruido de fondo de un determinado sistema.

Peso Sináptico: Es un peso determinado entre las conexiones de neuronas.

La media geométrica de una cantidad finita de números (digamos  $n$  números) es la raíz  $n$ -ésima del producto de todos los números.

$$\bar{x} = \sqrt[n]{\prod_{i=1}^n x_i} = \sqrt[n]{x_1 \cdot x_2 \cdot \dots \cdot x_n}$$

Peso de activación inicial: Peso dado inicialmente a cada neurona.

Umbral de excitación: es un peso asociado con la función de salida.

## ANALYTICAL SUMMARY

### HYBRID BETWEEN A NEURONAL NETWORK AND ARIMA MODEL (\*)

The present article is the result of a year of investigation with the intention of satisfying the Fasescoldas's need to present/display the commercial prices of the vehicles. This data is used by the insurance agencies in Colombia to fit their premiums on the commercial prices of the vehicles. This way those commercial prices, satisfy the client and generate an economic balance in case of any insurance claims.

By the characteristics of the problem a Neuronal Network was designed. Then a model that solved the difficulty that was appearing, to educate its brain and to begin to throw results, was the success with which it was managed to satisfy this necessity. Nevertheless appeared cases, in which there was not enough information feed the network. It was necessary to go to apply a model ARIMA on the resulting information and/or to use a smoothing index for the estimation of the price of a vehicle model that exists in the market, but not in the actual neurons. In these conditions the Hybrid Model was created.

The neuronal network consists of five neurons that are permanently educated (data bases of Colserauto (\*), more than 10.000 monthly clients at national level). An equal weight was initially assigned to them using the data bases of Colserauto. With the pertinent information as origin of the client price and source, the mean and median price or more frequent or wished price is found. Entering the weight that each neuron has applied to the transference function (as known value); the model is obtaining by itself the signal with significant information and the difference observed between the wished price and this signal. For the calculation of the synaptic weight, this difference is multiplied by the expected rate of geometric growth and the weight of initial activation. The smoothing factor is calculated over the synaptic weights. In case of a non existing vehicle model, this last factor is applied and the series is smoothed if there is not enough information. Otherwise a model ARIMA is applied for its estimation. If the concerned vehicle is one of the high range, a threshold ( $\emptyset$ ) is applied on the output function:  $Y_t = S(t) + S(t) * \emptyset$  and keeps the same conditions. This threshold is basically the monthly dollar exchange rate reference to another one.

### KEY WORDS

Artificial neuron (RNA): It is a learning paradigm and an automatic processing inspired by the form in which the nervous system of the animals works. It consists of simulating the properties observed in the biological neuronal systems through mathematical models created by means of artificial mechanisms.

Operation of the neuronal network: It is made up of units called neurons. Each neuron receives a series of entrances through interconnections and emits an exit. This exit comes given by a function from propagation or excitation, that generally consists of the addition of each entrance multiplied by weight of its interconnection. If the weight is positive, the connection denominates excitatory; if it is negative, inhibitory.

Hybrid: It is called Hybrid, the animal or the vegetable procreated by two individuals of different species or race. In this case the networks and the ARIMA model, look for an objective of estimation but in different routes.

Function of Transference: It is a model that facilitates the study of a neuron. The entrances to the network appear with a vector. For this case, represents a single neuron that contains

an element.  $W$  continues representing the weights and the net result of the network addition. The total output is determined by the transference function, which can be a function linear or nonlinear of  $n$ , That is chosen depending on the specifications of the problem that the neuron must solve. Although the RNA are inspired by biological models, does not exist limitation to make modifications in the output functions, so there will be artificial models that have nothing to do with the characteristics of the biological system.

Signal with significant information: The *Signal to noise ratio* SNR or S/N is defined as the margin that there is between the reference level (significant information) and [noise](#) of bottom of a certain system.

Synaptic weight: It is a weight determined between the neuron connections.

The geometric mean of a finite amount of numbers (we say  $n$  numbers) are the  $n$ - root of the product of all the numbers.

$$\bar{x} = \sqrt[n]{\prod_{i=1}^n x_i} = \sqrt[n]{x_1 \cdot x_2 \cdot \dots \cdot x_n}$$

Weight of initial activation: Initial weight given to each neuron.

Threshold of excitation: it is a weight associated with the output function.

# DOCUMENTACIÓN DEL MODELO

## HIBRIDO ENTRE UNA RED NEURONAL Y UN MODELO ARIMA

### 1. INTRODUCCION

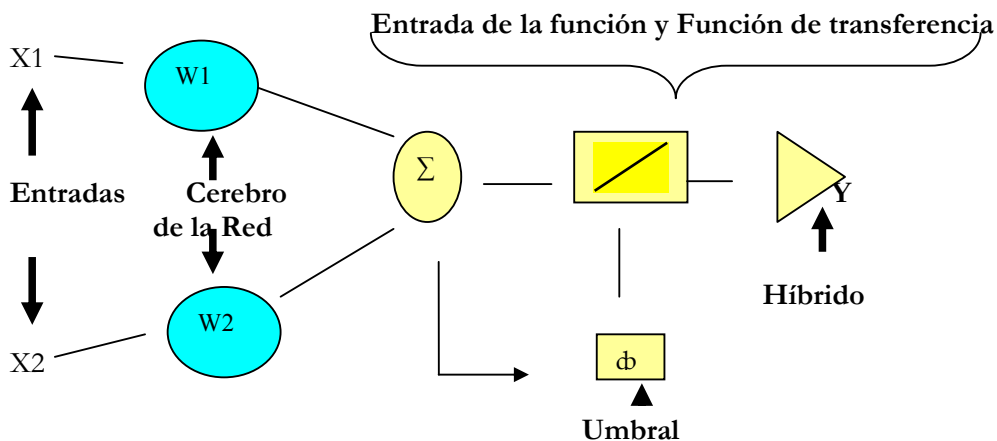
No puedo iniciar la introducción y desarrollo de la investigación sin dar especial agradecimiento al Ing. Pervys Rengifo, especialista en Inteligencia Artificial por sus orientaciones, a la paciencia de mi esposa e hijos por el tiempo regalado durante este trabajo y a la Universidades Militar y FUKL por su apoyo y colaboración.

Las **redes de neuronas artificiales (RNA)** son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Consiste en simular las propiedades observadas en los sistemas neuronales biológicos a través de modelos matemáticos creados mediante mecanismos artificiales (como un circuito integrado, un programa de computadora o un conjunto de válvulas). El objetivo es conseguir que las máquinas y/o programas den respuestas similares a las que es capaz el cerebro que se caracterizan por su generalización y su robustez.

Para el funcionamiento, una red neuronal se compone de unidades llamadas neuronas. Cada neurona recibe una serie de entradas a través de interconexiones y emite una salida. Esta salida viene dada por dos funciones:

- 1) Una función de propagación (también conocida como función de excitación), que por lo general consiste en la sumatoria de cada entrada multiplicada por el peso de su interconexión. Si el peso es positivo, la conexión se denomina excitatoria; si es negativo, se denomina inhibitoria.
- 2) Una función de activación, que modifica a la anterior. Puede no existir, siendo la salida la misma función de propagación.

Gráficamente un perceptrón sencillo de dos entradas para este híbrido se vería como:



## 2. METODOLOGÍA DEL MODELO: DISEÑO Y ESTRUCTURA

La aproximación basada en esta RNA parte de un conjunto de datos de entrada suficientemente significativo (5 bases de datos: Revista Motor, Otras revistas, Concesionarios, Retro alimentación Colserauto (Fasecolda- y/o información cliente), Clasificados, el objetivo es educar el cerebro de la red y en este sentido que *aprenda* las propiedades deseadas. De esta forma, el diseño de la red tiene menos que ver con cuestiones como los flujos de datos y la detección de condiciones, y más que ver con cuestiones tales como la selección del modelo de red, la de las variables a incorporar y el preprocesamiento de la información que formará el *conjunto de entrenamiento*. Asimismo, el proceso por el que los parámetros de la red se adecuan a la resolución de cada problema no se denomina genéricamente programación sino que se suele denominar *entrenamiento*.

La RNA diseñada para Fasecolda, se entrena alimentando las bases de datos y educando el cerebro de la misma.

Biológicamente, la estructura de un cerebro aprende mediante la reorganización de las conexiones sinápticas entre las neuronas que lo componen. De la misma manera, la RNA tienen un gran número de *procesadores* virtuales interconectados que de forma simplificada simulan la funcionalidad de las neuronas biológicas. En esta simulación, la reorganización de las conexiones sinápticas biológicas se modela mediante un mecanismo de **pesos**, que son ajustados durante la fase de aprendizaje. En la RNA entrenada, el conjunto de los pesos determina el *conocimiento* de esa RNA y tiene la propiedad de resolver el problema para el que la RNA ha sido entrenada, para nuestro caso cada base de datos se inició con igual peso (20%) para la información de cada una, por estudio de mercados y fuentes de información, los pesos se han venido cambiando y ajustando hasta alcanzar una salida con que el cliente y la Compañía de Seguros ha estado de acuerdo.

Por otra parte, en la RNA, además de los pesos y las conexiones, se tiene asociada una función matemática denominada función de transferencia. Dicha función genera la señal de salida a partir de las señales de entrada. La entrada de la función es la suma de todas las señales de entrada por el peso asociado a la conexión de entrada de la señal. Se adiciona un Umbral que se aplica en caso de un vehículo de gama alta. Algunos ejemplos de funciones de transferencia son la función escalón, la lineal o mixta, la sigmoideal y la función gaussiana. La Guía de Valores tiene asociada una función lineal o mixta.

Obtenidos los valores comerciales, bajo el entrenamiento y salida que nos ofrece la función, se aplica un ARIMA y/o un suavizamiento exponencial para la estimación de los valores de modelos inexistentes.

## 3. DESCRIPCIÓN DE LA RNA

Un **perceptrón** se refiere a una neurona artificial y también como a la unidad básica de inferencia en forma de discriminador lineal, que suele formar parte de una RNA.

Un perceptrón puede clasificar datos que sean linealmente separables. En el caso de un perceptrón con cinco entradas que es nuestro caso, deberá poder trazarse una única línea que separe las cinco clases que permite identificar el perceptrón.

Las funciones AND y OR son linealmente separables y por lo tanto pueden ser aprendidas por un perceptrón

La Topología de las RNA, se puede clasificar en función del patrón de conexiones que presenta. Así se definen tres tipos básicos de redes:

Dos tipos de redes de propagación hacia delante o acíclicas en las que todas las señales van desde la capa de entrada hacia la salida sin existir ciclos, ni conexiones entre neuronas de la misma capa.

- Monocapa. Ejemplos: perceptrón simple, Adaline. (Es una de las estructuras neuronales más simples junto con el perceptrón. Está formada por un único elemento de procesado (Adaline: Adaptive Linear Elemento; "Widrow-Hoff"), la estructura básica propuesta por Widrow-Hoff durante los Noventa son la base de la RNA que se aplicó para esta Red, el parte hecho por el autor consiste en la consecución de los pesos para el cerebro de la Red.
- Multicapa. Ejemplos: perceptron multicapa.
- Las redes recurrentes que presentan al menos un ciclo cerrado de activación neuronal. Ejemplos: Elman, Hopfield, maquina de Bolzman.

#### 4. RETROALIMENTACIÓN Y FUNCIONAMIENTO DEL MODELO

Quincenalmente se realiza un control de calidad sobre el perceptrón. Se seleccionan en forma aleatoria 100 códigos y se revisan sus valores versus las fuentes de alimentación.

Con base en los estudios de mercadeo que a diario se hacen, en la consulta precio cliente y clasificados, se argumentan los pesos que en el preceptrón alientan su cerebro, adicional a ello, hay un umbral que influye sobre la variación, que es la TRM y su variación mensual, se aplica la función lineal que conjuga las base de datos y este umbral, obteniéndose los valores comerciales por modelo.

Para los modelos de vehículos inexistentes o que no nos ofrece ninguna de las bases de datos pero que existen en el mercado, se aplica un modelo ARIMA y/o un suavizamiento exponencial, el cual sirve para hallar el valor comercial del modelo buscado.

Previo a la explicación rigurosa del modelo (punto 6), veamos como aplica la regla de aprendizaje, en la que se aplican los siguientes conceptos, que más adelante se explican:

$$S(t) = \sum_j w_{ij} o_j(t)$$

**S(t)** ; Señala emitida por información significativa

**w<sub>ij</sub>** : Pesos obtenidos mediante estudio de mercados de más de 10.000 vehículos mensuales

**o<sub>i</sub>** : es la señal producida por la unidad i (la unidad de salida)

La solución del siguiente caso se aplica, sobre cada modelo y sobre cada vehículo:

El cálculo de los pesos se hace mediante un estudio de mercados con base en la fuente de información que utiliza el dueño del vehículo y/o cliente Colserauto para avaluar su vehículo. Este estudio es permanente, dada la necesidad de alimentar la Red en forma actualizada.

Caso del cálculo para un Sprint modelo 1996, valores en millones de pesos

oj: Revista Motor, ok: Otras Revistas, ot: Clasificados, ou: Cliente y/o Fasecolda

ol: Concesionarios

valores:  $O(t)$  oj: 12.2 ok: 14.4 ot: 12 ou: 12,7 ol:13.2

pesos:  $(W_{ij})$  wil: 0.6 wik: 0.25 wit: 0.05 wiu: 0.05 wiul:0.05

$S(t)=12.8$  Señal emitida por variación con información significativa.

Para su retroalimentación, utilizamos  $T_i$ :

$T_i$  : Es el valor más deseado de la unidad  $i$  (la unidad de salida), hallado como la moda y/o mediana del estudio de mercados. Se utiliza para medir la diferencia con la Señal emitida con información significativa, para el caso se tiene.

$\Delta$ ; Es la variación entre modelos de la señal de salida, se utiliza para el cálculo de la media geométrica.

$ai(t)$ = activación inicial, todas las bases de datos o señales de salida, poseen el mismo peso al ingresar a la Red.

$\Delta w_{ij} = \sqrt[n]{\Delta}(T_i - S(t))a_i(t)$  Peso Sináptico.

$\alpha = \Sigma \Delta w_{ij} / n$  factor de suavizamiento esperado.

$\emptyset$ = Umbral, Variación tasa de crecimiento de TRM últimos dos meses.

$Y(t) = S(t) + S(t) * \emptyset$  Potencial sináptico para el caso de un vehículo alta gama

	Modelo								
	2004	2003	2002	2001	2000	1999	1998	1997	1996
$T_i$	22.4	20.7	19.5	18	16.9	15.8	14.8	13.9	12.9
$S(t)$	22.2	20.5	19.2	17.9	16.7	15.6	14.6	13.5	12.8
$T_i - S(t)$	0.2	0.2	0.3	0.1	0.2	0.2	0.2	0.4	0.1
$\Delta$		1.08	1.07	1.07	1.07	1.07	1.07	1.09	1.05
$\sqrt[n]{\Delta}$	1.07								
$\Delta w_{ij}$	0.05	0.04	0.08	0.02	0.04	0.05	0.04	0.11	0.02
$\alpha$	0.04								

Es decir un Sprint, Modelo 1996 se consigue en 12.8 millones.

Aplicamos el ARIMA y/ aplicamos  $\alpha$  como factor de suavizamiento exponencial, con los valores hallados para cada uno de los modelos y calcular un modelo al que se le desconoce le valor.

## 5. BUSQUEDA DE LA CALIDAD TOTAL

En la búsqueda de la calidad total, para el ajuste de precios por modelo de cada vehículo, continuamente se esta en contacto con concesionarios especializados y compraventas de vehículos, esto ha repercutido en el mejoramiento de la calidad de la información de los vehículos y en la supervisión de sus valores.

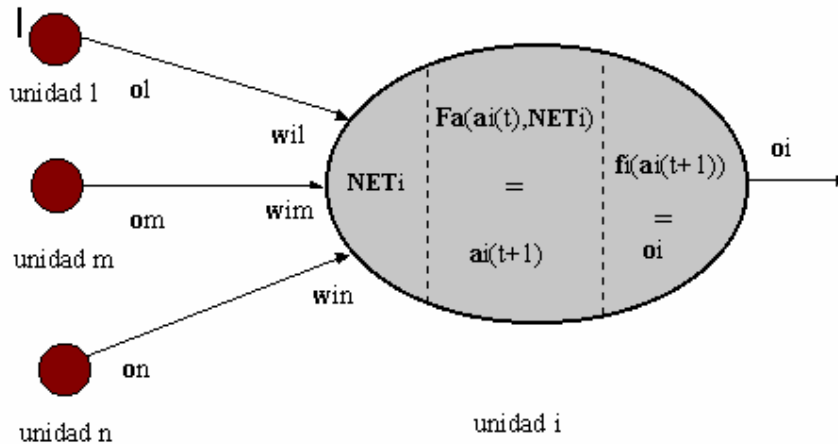
Colserauto ha estado contratando un programador para mejorar la calidad y prestancia de la información de la Red Neuronal, se espera para las próximas guías poder mostrar su desarrollo bajo el diseño aplicado.

## 6. EXPLICACION DEL FUNCIONAMIENTO

La tarea de cada neurona o unidad es simple: **recibir las entradas de las células vecinas y calcular un valor de salida, el cual es enviado a todas las células restantes.** Toda la información que la red recibe, almacena, transforma y envía se expresa *en términos matemáticos*, en la forma de cantidades y de fórmulas aplicadas a cantidades.

La representación interna de la información se genera en las sinapsis o conexiones y en la actividad de las neuronas, particularmente en las de la capa oculta. Dado que las redes modifican sus estados a través del tiempo como consecuencia de las distintas entradas de información, es imprescindible también introducir en la descripción de los estados de activación la referencia al tiempo:

1. a cada neurona (unidad  $i$ ,  $u_i$ ) le corresponde en cualquier instante un valor numérico denominado *valor o estado de activación*  $a_i(t)$ ;
2. cada unidad utiliza una *función de salida*,  $f_i$ , que calcula la señal de salida (“ $o_i$ ”, output ) a partir del estado de activación que en ese momento tiene la neurona;
3. la señal de salida se trasmite a otras unidades con las que dicha neurona está conectada;
4. una neurona recibe la señal de todas las unidades con las que está conectada, pero lo hace teniendo en cuenta:
  - el peso asociado a cada una de las conexiones,  $w_{ij}$ .
  - y la intensidad o señal que se transmite por la sinapsis; a la suma de todas las señales ponderadas por el peso de las correspondientes conexiones se le llama **NET<sub>i</sub>**
5. el nuevo estado de activación  $a_i(t+1)$  de la neurona  $i$  se calcula mediante una función de activación **F<sub>a</sub>**, teniendo en cuenta la entrada total calculada o **NET** y su anterior estado de activación,  $a_i(t)$



(Tomado de "Redes neuronales artificiales", J. R. Hilera y V. J. Martínez, Editorial Rama; p. 51.)

La **notación empleada** en las redes neuronales no es unánime:

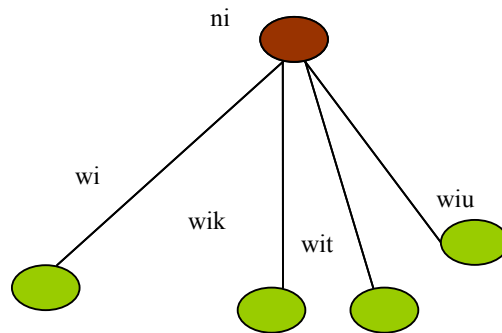
- las **unidades o neuronas** se representan como "u" o como "n"
- la **función de activación** se suele representar por Fa, pero también, simplemente, por F; dado que es preciso referirse a dos funciones, la de activación y la de transferencia, conviene utilizar una notación distinta para cada una de ellas, por ejemplo para la **función de activación** la F mayúscula (**Fa**) y para la de transferencia la f minúscula (**fi**);
- la referencia a la **activación actual** se suele representar como a(t) y la **siguiente** como a(t+1); pero en la literatura sobre el tema también podemos encontrar la referencia a la activación actual como a(t-1) y el estado de activación siguiente como a(t) (y tanto en negrita como en caracteres normales);
- para distinguir la parte de la fórmula que corresponde al **nombre de las unidades** de la parte de la fórmula correspondiente a los otros elementos, se pueden emplear varias estrategias:
  - subíndice para los nombres y letra normal para el resto:  $o_i$
  - letra normal para el nombre de la unidad y negrita para el resto:  $o_i$
- para referirse al **peso** de cada conexión se puede emplear tanto la O mayúscula como la o minúscula, lo mismo que para el peso ( $W$  o  $w$ ) y para la **entrada total** en cada unidad ( $Net$  o  $NET$  O  $NET$  o  $net$ ).

## LA ENTRADA (INPUT) TOTAL. LA REGLA DE PROPAGACIÓN

Una neurona recibe señales desde múltiples neuronas por lo que es necesario calcular el efecto global que tendrá ese conjunto de señales o inputs. El input total se denomina **NET**, y para el caso de las entradas que le llega a la neurona i en el tiempo t, **NET<sub>i</sub>(t)**. Llamamos **regla de propagación** a la fórmula que nos indica cómo calcular dicha entrada total o NET; existen varias reglas de propagación, pero la más sencilla y habitual es la que se representa del siguiente modo:

$$NET_{i(t+1)} = \sum_j w_{ij} * o_j(t)$$

$NET_i$  es la suma de todas las señales que llegan a la neurona  $i$  (señales representadas como  $o_j$ ), ponderada cada una de ellas por el peso de conexión asociado, ( $w_{ij}$ ).



Los valores que transmite una neurona a las neuronas con las que está conectada se representan como “ $o$ ” ( $o_j$  sería el valor que transmite la neurona  $j$ ). Pongamos que los valores que transmite las neuronas de la capa de entrada, y los pesos correspondientes a sus conexiones con la neurona  $i$  son los siguientes:

Los valores corresponden a un Sprint modelo 96, donde:

$o_j$ : Revista Motor,  $o_k$ : Otras Revistas,  $o_t$ : Clasificados,  $o_u$ : Cliente y/o Fasecolda

$o_l$ : Concesionarios

valores:  $O(t)$   $o_j$ : 12.2       $o_k$ : 14.4       $o_t$ : 12       $o_u$ : 12,7       $o_l$ :13.2

pesos: ( $W_{ij}$ )  $w_{il}$ : 0.6       $w_{ik}$ : 0.25       $w_{it}$ : 0.05       $w_{iu}$ : 0.05       $w_{il}$ :0.05

En ocasiones los pesos de todas las conexiones o sinapsis correspondientes a una neurona se suelen representar mediante una matriz: la matriz  $W_{ij}$  incluiría todos los pesos de las conexiones con la neurona  $i$  ( $w_{i1}, w_{i2}, w_{i3}, \dots, w_{in}$ ); si un elemento de la matriz es positivo indica que la conexión es excitatoria, si es negativo, la conexión es inhibitoria y si es 0 no existe conexión entre neuronas; en el caso anterior la matriz sería:

$$W_{ij} = [0.6, 0.25, 0.05, 0.05, 0.05]$$

También es habitual referirse al conjunto de valores de entrada de una unidad como el vector  $O(t)$  (de “output”, salida, puesto que normalmente estos valores que le llegan a una unidad son la salida que produce otra unidad). En el caso anterior, el vector de valores que le llegan desde otras neuronas y correspondiente al instante anterior a la modificación de la neurona  $i$  (instante que se suele representar como  $(t)$ ) sería:

$$O(t) = [12.2, 14.4, 12, 12.7, 13.2]$$

Si utilizamos la regla de propagación antes señalada, la entrada total que le llega a la neurona  $i$  o  $\text{NET}_i$  será la suma de las entradas parciales ponderadas por los pesos correspondientes:

$$\text{NET}_i = \mathbf{S}(\mathbf{t}) = \sum_j w_{ij} o_j(t) \\ (12.2 \times 0.6) + (14.4 \times 0.25) + (12 \times 0.05) + (12.7 \times 0.05) + (13.1 \times 0.05) = 12.8$$

La entrada total se calcula para después averiguar el estado de activación que la va a corresponder a la unidad, pero, dado que la regla de propagación citada es muy sencilla, la función de activación que ha de calcular su activación no se suele referir a  $\text{NET}_i$ , sino simplemente al sumatorio de los productos del peso por el valor de las entradas ( $W_{ij} * O_j$ ).

## EL ESTADO DE ACTIVACIÓN

La **activación** (o **excitación**, o **actividad**) de una neurona es muy importante para el procesamiento puesto que la salida que la neurona provoca depende de ella. La activación es el **grado o nivel de excitación de una neurona**, y, generalmente, cambia en función de cuatro factores:

- el anterior estado de activación de la unidad;
- las entradas que la neurona recibe;
- los pesos de las conexiones por las que recibe las señales;
- la función de activación utilizada para calcular la activación a partir de dichas entradas.

Se llama **ciclo** a cada una de las veces que la red toma los vectores de entrada (conjunto de valores de entrada) para realizar los cálculos correspondientes; pues bien, dado que en cada ciclo la activación de las neuronas puede ser distinta, es preciso caracterizar dicha activación en cada pasada de la información o ciclo, por ello es necesario referirse al tiempo; así, es habitual representar como  $\mathbf{A}_i(t)$  (o también, con minúscula, como  $\mathbf{a}_i(t)$ ), el nivel de activación de la neurona  $i$  en el tiempo  $t$ .

Lo que la red representa en cada momento depende del conjunto o **patrón de activación** de la totalidad de las unidades que la componen, por ello es conveniente también especificar dicho patrón. Para reflejar el patrón de activación se suele utilizar un vector de  $N$  números reales  $\mathbf{a}(t)$ , vector que representa el estado de activación de todas las unidades de la red en el tiempo  $t$ ; en el vector, cada uno de los elementos representa la activación de cada unidad en el tiempo  $t$ :

$$\mathbf{A}(t) = (a_1(t), a_2(t), \dots, a_i(t), \dots, a_n(t))$$

De este modo, cabe caracterizar el procesamiento que realiza la red como la evolución que sufren a través del tiempo los patrones de activación de las unidades.

Hay muchos modelos de redes conexionistas y los valores de activación que pueden tomar las unidades no son los mismos en todos ellos. Los valores utilizados en nuestra Red son:

## CONTINUOS

- \* **abiertos**; los modelos que utilizan este tipo reciben el nombre de *modelos no acotados* (o modelos sin límite): el valor de activación de una neurona  $i$  puede ser cualquier número real;

Naturalmente, el comportamiento y las utilidades de las redes con distintos tipos de valores de activación son también distintos; por ejemplo, si las entradas responden a información analógica y no a señales discretas, entonces los valores más adecuados son los continuos.

## LA REGLA DE ACTIVACIÓN

### CONCEPTO Y FÓRMULA GENERAL

El estado de activación de una unidad depende de la **función de activación**, que no es otra cosa que una *función matemática utilizada para calcular el siguiente estado de activación*; dicha función calcula la nueva actividad de la unidad o neurona a partir de:

- el anterior estado de activación de la unidad (aunque esta cantidad no siempre se tiene en cuenta);
- todas las entradas que recibe dicha unidad de las otras unidades;
- los pesos de las conexiones que la unidad en cuestión tiene con las unidades de las que recibe las entradas.

Si tenemos la unidad  $i$ , su actividad en el tiempo  $t+1$  dependerá de la función de activación (**Fa**) aplicada a:

- el estado de activación de la neurona  $i$  en el tiempo  $t$ ;
- todas las entradas  $o_j$  recibidas por la unidad  $i$  desde las neuronas  $j$  de la capa anterior en el tiempo  $t$ ;
- los pesos de conexión  $w_{ij}$  correspondientes a las distintas sinapsis por los que cada señal  $o_j$  llega a la neurona  $i$  en el tiempo  $t$ .

De este modo, podemos representar la función de activación para el cálculo del nuevo estado de activación de la unidad  $i$  en el tiempo  $t+1$  utilizando la siguiente notación:

$$a_i(t+1) = \mathbf{Fa} [a_i(t), o_j(t), w_{ij}]$$

Dado que la entrada total o NET correspondiente a dicha unidad es igual a la suma de todas las entradas ponderadas por sus correspondientes pesos, la anterior fórmula también se puede expresar del siguiente modo:

$$a_i(t+1) = \mathbf{Fa} [a_i(t), \mathbf{NET}_i(t)]$$

Como más arriba se ha indicado, la notación utilizada no es la misma en todos los autores, así en la fórmula algunos utilizan **F**, otros **Fa**, y para referirse al tiempo algunos utilizan la notación  $t$  y  $t-1$  y otros  $t+1$  y  $t$ , pero son fórmulas equivalentes.

Como ocurría para los valores de activación de cada neurona, existen múltiples funciones de activación, y es el diseñador de la red quien debe establecer la que se ha de utilizar en función de las peculiaridades de la tarea para la que se diseña la red.

## **FUNCIONES DE ACTIVACIÓN**

- ***FUNCIÓN IDENTIDAD***

Se trata de una función muy simple que no tiene en cuenta el anterior estado de activación de la unidad. Se puede expresar con la siguiente fórmula:

$$a_i(t+1) = \sum_j w_{ij} o_j(t)$$

Según esta función de activación, el estado de activación que le corresponde a la unidad  $i$  ( $a_i(t+1)$ ) tras recibir las señales desde las unidades a las que está conectada es el resultado de sumar todas las señales recibidas ( $o_j$ ), ponderadas cada una de ellas por el peso sináptico ( $w_{ij}$ ) asociado a la sinapsis por la que se transmite la señal.

Podemos representar el conjunto de estímulos de entrada que le llegan a una unidad desde la capa anterior como un vector, y los pesos sinápticos correspondientes mediante una matriz; en esta circunstancia, el cálculo del nuevo estado de activación de todas las neuronas se obtiene simplemente mediante el producto de matrices.

En algunas redes se incluye en la función de activación un factor que representa una fracción del valor de activación anterior, por ejemplo:

$$a_i(t+1) = \frac{1}{2} a_i(t) + \sum_j w_{ij} o_j(t)$$

Con esta variante conseguimos que la unidad o neurona mantenga cierto estado de activación durante varios ciclos, incluso en el caso de que las entradas sean cero, en esta última circunstancia en cada período de tiempo su estado de activación corresponderá a la mitad del valor de activación anterior. Con este tipo de funciones de activación se garantiza que cada neurona no varíe en su estado y en su respuesta demasiado bruscamente (recordamos que la respuesta de la red depende en gran medida de los estados de activación de las unidades que la componen).

## **FUNCIÓN DE SALIDA O DE TRANSFERENCIA**

Las funciones mixta y sigmoideal son las más apropiadas cuando queremos como salida información analógica.

- ***FUNCIÓN IDENTIDAD O FUNCIÓN LINEAL***

Es la más sencilla y la de menos uso. Se llama función de identidad *porque la salida correspondiente a la unidad es igual a su estado de activación*. En realidad esta función equivale a no aplicar función de salida. La fórmula correspondiente a esta función es

$$f_i[a_i(t+1)] = a_i(t+1)$$

y la salida correspondiente a dicha unidad se puede representar como

$$o_i(t+1) = a_i(t+1)$$

La función de identidad es adecuada cuando la función de activación que hemos utilizado para calcular la activación de la unidad es de tipo umbral. En este caso la activación de la unidad será 1 si está activa y 0 si está inactiva, y la salida correspondiente será 1 en el primer caso y 0 en el segundo (o 1 y -1 si hemos preferido esta notación).

Atendiendo a los tipos de funciones que intervienen en el cómputo, las neuronas se pueden clasificar en neuronas lineales y neuronas no lineales.

- ***Neuronas lineales***

Tienen funciones de activación y de transferencia lineales, por lo que la composición de ambas funciones da lugar a otra función lineal. En estas neuronas la salida  $o_i$  es linealmente dependiente (es decir, proporcional) de sus entradas  $O_j$  ponderadas cada una de ellas por el peso asociado a las sinapsis ( $W_{ij}$ ) por las que le llega a la unidad  $i$  las entradas señal. La respuesta de las neuronas lineales no está acotada y puede tomar cualquier valor.

Está demostrado que los cómputos que realizan redes con más de dos capas con unidades lineales se pueden realizar también con redes de dos capas, con lo que se hace superfluo construir redes de más de dos capas si las funciones utilizadas son siempre lineales.

Las primeras redes que aparecieron eran de este tipo.

Existen varios modelos de aprendizaje, siendo los principales el supervisado, el no supervisado y el reforzado.

- **Aprendizaje supervisado:**

En este modelo existe un agente externo (supervisor o maestro) que controla el proceso de aprendizaje de la red. La red debe relacionar dos fenómenos ( $X$  e  $Y$ ) mediante la presentación de un conjunto de ejemplos  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ . En la formulación anterior, " $X$ " representa las entradas e " $Y$ " las salidas requeridas. Se llama "*juego de ensayo*" al conjunto formado por las parejas anteriores, "*patrón de estímulos-respuesta deseada*". En algunos casos no hay ninguna duda en cuanto a los elementos que deben componer dicho juego de ensayo puesto que se conocen perfectamente todos los patrones que la red debe reconocer y sus salidas correspondientes (este ocurre, por ejemplo con el uso del Perceptrón para el cómputo de las funciones lógicas); pero en la mayoría de los casos esto no es así y es preciso tener mucho cuidado en la elección del juego de ensayo y tratar de incluir en él los patrones más representativos del problema o fenómeno que se intenta computar. El maestro presenta a la red una entrada  $x$  y la red produce una salida  $o_i$ . Normalmente esta salida no coincide con la salida requerida, por lo que el maestro debe calcular el *error de salida*,  $e_i = e(o_i, y_i)$  y proceder a la modificación de los pesos utilizando alguna regla de aprendizaje con la intención de aproximar la salida obtenida a la salida deseada. Tras esta modificación se presenta el siguiente patrón del juego de ensayo y se procede de la misma manera. Cuando se termina con el último patrón del juego de ensayo, se tiene que volver a empezar de nuevo pues los pesos se han modificado y es preciso comprobar que la red responde adecuadamente. A cada uno de los pasos completos del juego de ensayo se le llama *ciclo*; dependiendo de la complejidad del problema, serán precisos pocos o muchos ciclos para el aprendizaje. La fase de aprendizaje termina cuando los pesos se *estabilizan o convergen* en unos valores óptimos. En este caso la red consigue responder correctamente a todas las presentaciones de los patrones estimuladores del juego de ensayo. Aunque el ideal de aprendizaje es el cien por cien de aciertos, se considera que

concluye cuando se minimizan razonablemente los errores ante el conjunto de ejemplos presentados.

### **REGLA DE WINDROW-HOFF O REGLA DELTA O REGLA DEL MENOR ERROR CUADRÁTICO**

La utilizaron por primera vez, en 1960, B. *Widrow* y M. *Hoff* para su red ADALINE. Se trata de una regla de aprendizaje supervisado (es preciso presentar a la red todos los casos de pares entrada-salida que se desea que la red aprenda, y que el maestro modifique los pesos de las conexiones para corregir los errores en las salidas). Se denomina “regla delta” porque en esencia el procedimiento consiste en modificar los pesos de las conexiones (por lo tanto, favorecer el aprendizaje de la red) a partir del cálculo de la *diferencia* (o delta) entre las salidas efectivas de la red y las salidas deseadas.

El maestro presenta a la red un vector o patrón de entrada. Si la red produce el vector de salida deseado no realiza ningún cambio en los pesos. Si existe discrepancia entre ambos vectores, los pesos se cambian para reducir la diferencia. Cuando existe diferencia entre las salidas efectivas y las deseadas la modificación afecta a todos los pesos de las neuronas que colaboran en el error, y se modifican en proporción a su influencia en el error. Se sigue el procedimiento citado minimizar el error y, en los mejores casos, convertirlo en cero. En la literatura sobre el tema existen varias formulaciones; presentamos dos de las más sencillas:

$$\Delta w_{ij} = \mu [T_i(t+1) - a_i(t+1)] o_j(t)$$

$\Delta w_{ij}$  : incremento o decremento del peso sináptico

$\mu$  : tasa de aprendizaje

$T_i(t+1)$  : activación deseada en la neurona  $i$

$a_i(t+1)$  : activación real de la neurona  $i$

$o_j(t)$  : señal emitida por la neurona  $j$  y enviada a la neurona  $i$

En esta primera fórmula intervienen los valores de los estados de activación (el que realmente tiene la unidad y el deseado), por lo que para averiguar si la salida final de la red es la correcta es preciso tener en cuenta la función de transferencia.

Otra formulación es la siguiente:

$$\Delta w_{ij} = \mu [T_i(t+1) - o_i(t+1)] a_j(t)$$

$\Delta w_{ij}$  : incremento o decremento del peso sináptico

$\mu$  : tasa de aprendizaje

$T_i$  : es la señal deseada de la unidad  $i$  (la unidad de salida)

$o_i$  : es la señal producida por la unidad  $i$  (la unidad de salida)

$a_j(t)$  : es la activación de la unidad  $j$  (la unidad de entrada)

La expresión entre corchetes representa el error que se comete en el nivel de activación de la neurona. El nuevo peso sináptico  $w_{ij}$  se calcula sumando o restando el incremento obtenido con la fórmula general.

## **Bibliografía**

Bernard **Widrow** e Marcian Hoff, [INTELLIGENZA ARTIFICIALE](http://www.google.com.co/search?hl=es&ie=ISO-8859-1&q=WIDROW+AND+HAFF&btnG=B%FAsqueda&meta=),

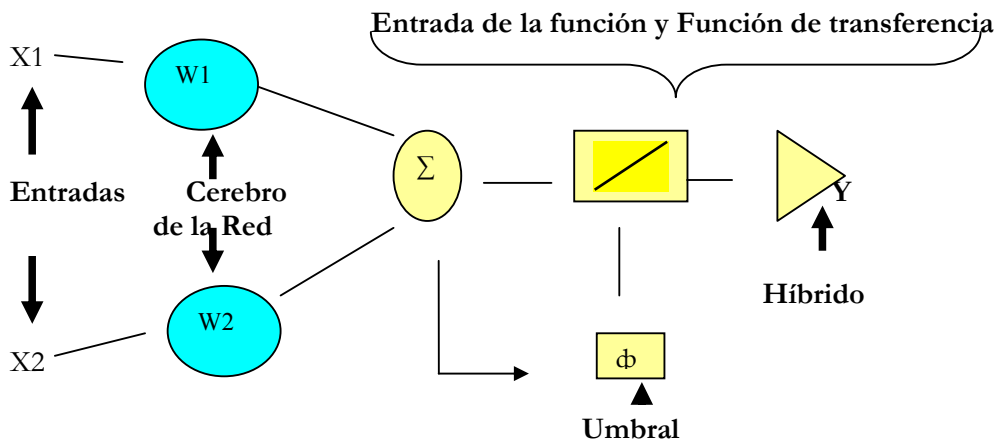
J. R. Hilera y V. J. Martínez , **Redes neuronales artificiales** , Editorial Ra-ma; p. 51.)

## GRAFICAS

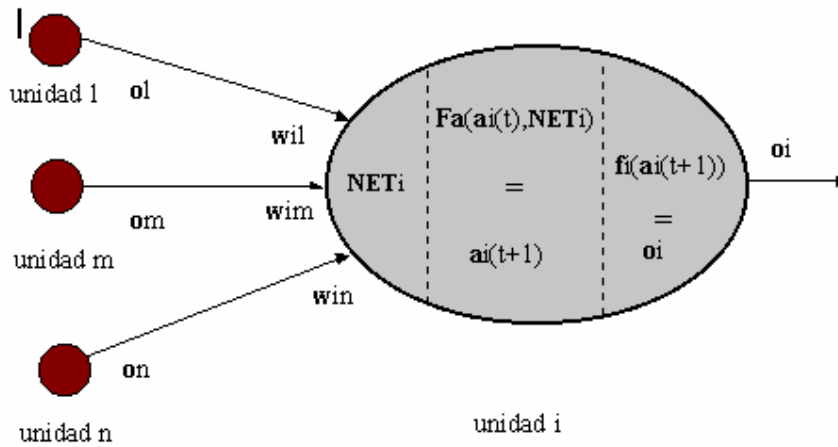
### Media Geométrica

$$\bar{x} = \sqrt[n]{\prod_{i=1}^n x_i} = \sqrt[n]{x_1 \cdot x_2 \cdot \dots \cdot x_n}$$

### Perceptrón



### Red Neuronal Artificial



## Señales presentadas como Oj

