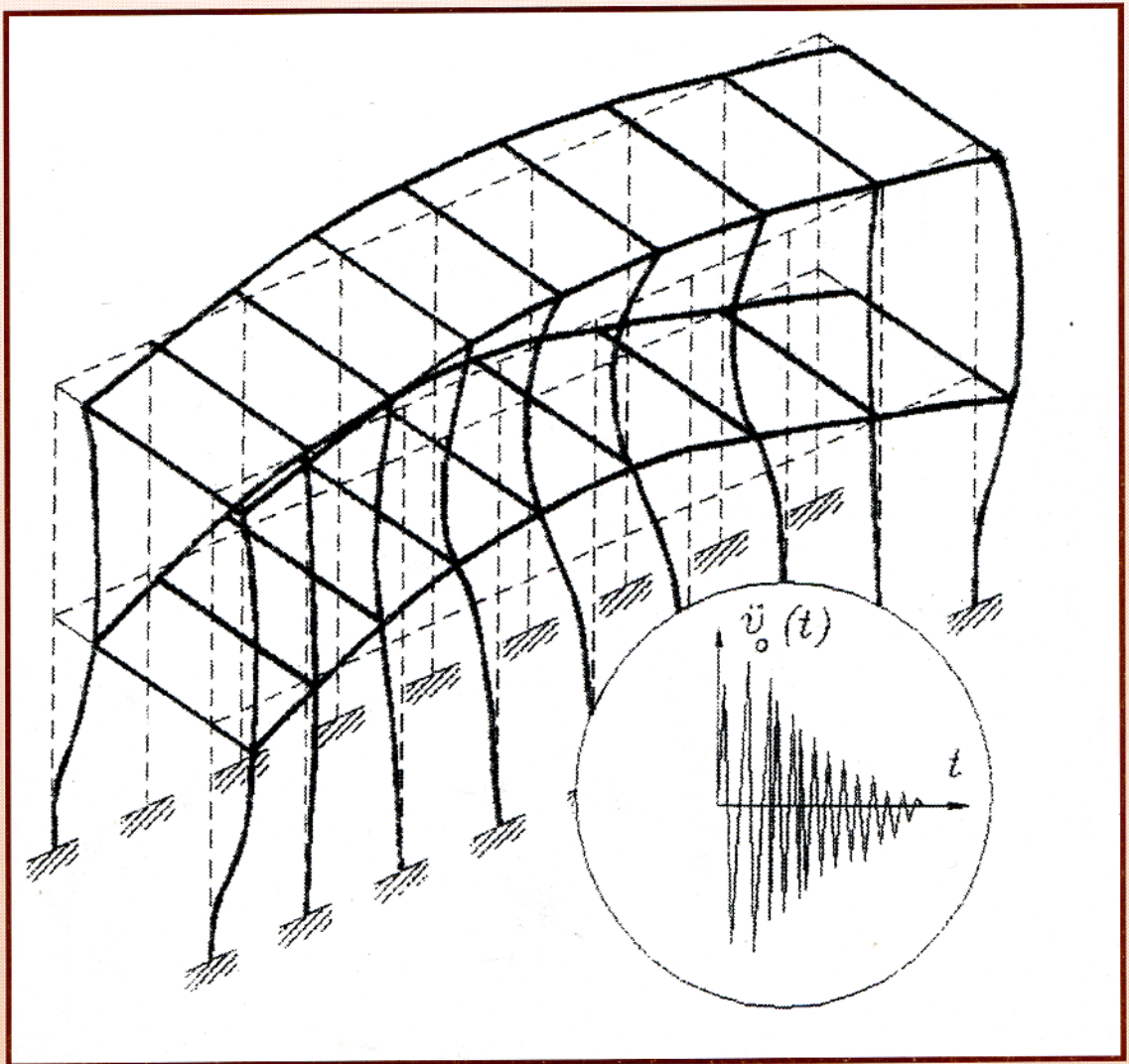


# Ingeniería

Revista de la Universidad de Costa Rica  
Julio/Diciembre 1997 VOLUMEN 7 Nº 2



# INGENIERIA

Revista Semestral de la Universidad de Costa Rica  
Volumen 7, Julio/Diciembre 1997 Número 2

## DIRECTOR

Rodolfo Herrera J.

## CONSEJO EDITORIAL

Víctor Hugo Chacón P.

Ismael Mazón G.

Domingo Riggioni C.

## CORRESPONDENCIA Y SUSCRIPCIONES

Editorial de la Universidad de Costa Rica  
Apartado Postal 75  
2060 Ciudad Universitaria Rodrigo Facio  
San José, Costa Rica

## CANJES

Universidad de Costa Rica  
Sistema de Bibliotecas, Documentación e Información  
Unidad de Selección y Aquisiciones-CANJE  
Ciudad Universitaria Rodrigo Facio  
San José, Costa Rica

### Suscripción anual:

Costa Rica: ₡ 1 000,00

Otros países: US \$ 25,00

### Número suelto:

Costa Rica: ₡ 750,00

Otros países: \$ 15,00





# PREDICCIÓN DE CAUDALES Y CONSUMOS EN ACUEDUCTOS MEDIANTE REDES NEURONALES DE PROPAGACIÓN HACIA ATRÁS

Juan José Ugalde-Chacón\*

## Resumen

Se muestra una aplicación comercial de redes neuronales: predicción de caudales y consumo de los abonados de un acueducto. Esta es la primera red neuronal desarrollada a nivel comercial e industrial en Costa Rica y su importancia radica, además de eso, en que constituye una herramienta poderosa para la planificación de nuevas estructuras de acueductos, dado que los datos proporcionados muestran las necesidades futuras de expansión, lo cual resulta sumamente adecuado para organizaciones en donde las nuevas obras deben planearse con mucha anticipación. La técnica utilizada es la de propagación hacia atrás ("Back Propagation"), la cual es aplicada para obtener diez redes neuronales. Como producto del análisis se determina la necesidad de una mayor experimentación en este campo, la forma en que deben manejarse los parámetros de la red (factor de aprendizaje y unidades de capa oculta principalmente) y la capacidad de cómputo ideal para este tipo de proyectos, así como los campos de aplicación para técnicas similares y la eficacia de este método para predicción de comportamientos de nacientes o de población.

## Summary

A commercial application of neural networks is shown, which consists of aqueduct consumers demand and production prediction. This is the first neural network developed on an industrial-commercial level in Costa Rica. It is important because it constitutes a powerful tool for planning new aqueduct structures, given that the output data show the future needs of expansion, which results extremely adequate for organizations where one must justify and plan new works with great anticipation. The technique used here is Back Propagation and it is implemented in order to develop ten networks. As a result of the analysis it is determined that more experimentation in this field is needed: how the adequate handling of the network parameters (learning factor and hidden units) must be done, the computation capacity for this kind of projects, as well as the application fields for similar techniques, and the efficiency of this method for predicting behaviors of springs or populations.

## 1. INTRODUCCIÓN

La filosofía de las redes neuronales se ubica dentro del área del control automático, utilizada para identificación de estados en sistemas y, en general, para aprendizaje de comportamientos de cualquier sistema del cual pueda obtenerse suficiente información.

La filosofía mencionada se basa en la teoría del conexionismo, la cual surgió por investigaciones en campos como la neurobiología.

### A. Estado actual de las redes neuronales

Actualmente este campo se encuentra en un estado de gran crecimiento, fomentado por su continuo desarrollo e investigación en todo el mundo. Se están utilizando en un amplio rango

a nivel de experimentación aunque comercialmente apenas se están consolidando.

De hecho, aún no existen guías apropiadas para el diseño de aplicaciones de redes específicas (los llamados *Cook books*), la bibliografía es muy general o, en su defecto, enfocada a casos muy aislados. Por tales condiciones se está en una etapa de pre-comercialización.

Muchas de las aplicaciones exploradas son militares, como reconocimiento y rastreo automático, control autónomo de vehículos y técnicas de reconocimiento de patrones para inteligencia militar. También se utilizan en otros campos fuera del militar, tales como la compresión de datos.

**B. En Costa Rica**

A pesar del desarrollo de las redes a nivel mundial, en nuestro país no han sido aplicadas a nivel industrial o comercial hasta el momento. No existen tampoco referencias locales de sistemas generales de redes neuronales como el utilizado en este proyecto para generar las redes.

En lo que respecta a sistemas de acueductos en Costa Rica, ningún método de predicción de caudales o consumos ha sido realizado, a pesar de que, como se demuestra en este artículo, es una excelente herramienta para planificación.

**C. Estructura del artículo**

El artículo se divide en cinco secciones, la que sigue es de trabajo teórico y expone los fundamentos de la técnica de propagación hacia atrás, además se describe el caso al cual se aplicó dicha técnica con su respectiva justificación, y las expectativas de respuesta de las redes basados en estimaciones teóricas; en la

tercera sección se exponen los resultados experimentales obtenidos al montar las redes en un computador personal.

La siguiente sección expone las conclusiones del trabajo según la experiencia cosechada y, finalmente, se proponen algunas posibles expansiones en el campo, específicamente en nuestro país, tomando como punto de partida los resultados del presente trabajo.

**1. MARCO TEÓRICO**

**A. Propagación hacia atrás**

Dentro del ámbito de las redes neuronales existen varios métodos de aprendizaje; el más popular y fuerte es el desarrollado por Rumelhart y McClelland [6] llamado "Back Propagation" o propagación hacia atrás, el cual es un método de aprendizaje supervisado.

En la siguiente figura se muestra el esquema básico de una red de este tipo.

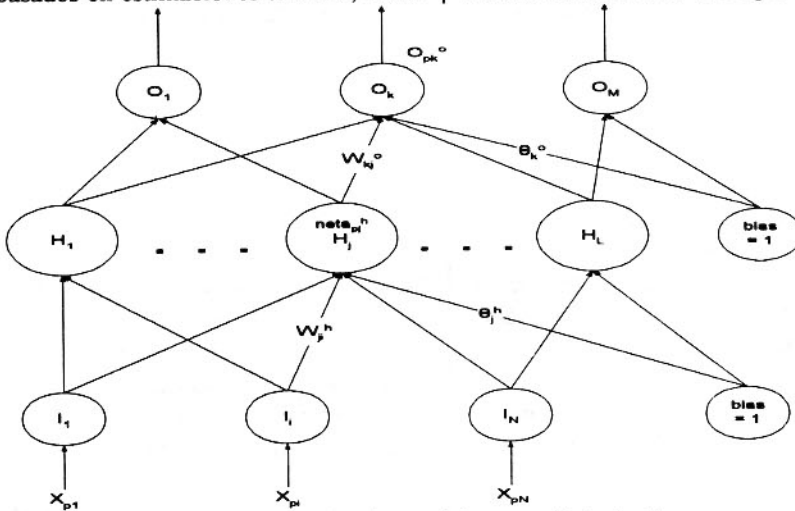


Ilustración 1 : Diagrama básico de una red de propagación hacia atrás.

La red consiste en tres capas: una de entradas con N unidades de proceso, una de salidas con M unidades de proceso y una capa oculta con L neuronas.

Terminología:

$x_{pi}$  es la i-ésima entrada del patrón p-ésimo.

$I_i$  es la unidad de proceso i-ésima de la capa de entradas.

$H_j$  es la unidad de proceso j-ésima de la capa oculta.

$O_k$  es la unidad de proceso k-ésima de la capa de salidas.

$w_{ji}^h$  es el peso de la conexión entre la salida de la neurona de entrada i-ésima y la j-ésima unidad oculta.

$\theta_j^h$  es el peso del bias que ingresa a la neurona j-ésima de la capa oculta.

$w_{kj}^o$  es el peso de la conexión entre la salida de la neurona oculta j-ésima y la k-ésima unidad de proceso de la capa de salidas.

$\theta_k^o$  es el peso del bias que ingresa a la neurona k-ésima de salida.

Se utiliza la regla delta generalizada (GDR) para sintetizar las ecuaciones de la red neuronal con propagación hacia atrás. La información se propaga hacia adelante por las capas.

### Procesamiento de los datos

La entrada neta a la j-ésima unidad de la capa oculta es:

$$neta_{pj}^h = \sum_{i=1}^N w_{ji}^h x_{pi} + \theta_j^h \quad (1)$$

La activación de ese nodo de la capa oculta es:

$$i_{pj} = f_j^h(neta_{pj}^h) \quad (2)$$

Para los nodos de salida son válidas las siguientes ecuaciones:

$$neta_{pk}^o = \sum_{j=1}^L w_{kj}^o i_{pj} + \theta_k^o \quad (3)$$

Además:

$$o_{pk} = f_k^o(neta_{pk}^o) \quad (4)$$

### Pasos para el entrenamiento de la red

Aplicar un vector de entradas y calcular las salidas.

Comparar con las salidas deseadas y determinar una medida del error.

Determinar en qué dirección y cuánto debe cambiar cada peso para reducir el error.

Aplicar las conexiones a los pesos.

Repetir los pasos 1 a 4 con todos los patrones, hasta que el error sea aceptable.

### Actualización de pesos

El error en cualquier k-ésima unidad de salida, evaluando el p-ésimo patrón es el siguiente:

$$\delta_{pk} = (y_{pk} - o_{pk}) \quad (5)$$

donde:  $y_{pk}$ : salida deseada (del patrón),

$o_{pk}$ : salida obtenida.

Se busca minimizar la suma de los cuadrados de los errores de todas las salidas, sea:

$$E_p = \frac{1}{2} \sum_{k=1}^M \delta_{pk}^2 \quad (6)$$

Se determina la dirección en que deben cambiar los pesos calculando el valor negativo del gradiente de  $E_p$  con respecto a los pesos  $w_{kj}$ . La magnitud del cambio en el peso es proporcional al gradiente negativo. Así, los pesos deben actualizarse de la siguiente manera:

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \Delta_p w_{kj}^o(t) \quad (7)$$

donde:

$$\Delta_p w_{kj}^o = \eta (y_{pk} - o_{pk}) f_k^{o'}(neta_{pk}^o) i_{pj} \quad (8)$$

$\eta$ : factor de velocidad de aprendizaje

### Función de salida

La función de salida puede ser de dos formas, a saber:

1. Función lineal:  $f_k^o(neta_{pk}^o) = neta_{pk}^o \Rightarrow f_k^{o'}(neta_{pk}^o) = f_k^{o'} = 1$

2. Función logística (sigmoide):

$$f_k^o(neta_{pk}^o) = (1 + e^{-neta_{pk}^o})^{-1} \Rightarrow f_k^{o'}(neta_{pk}^o) = f_k^{o'} = o_{pk}(1 - o_{pk})$$

El cambio de pesos, entonces dependerá de la función de salida elegida.

Definimos el término de error para las unidades de salida como:

$$\delta_{pk}^o = (y_{pk} - o_{pk}) f_k^{o'}(neta_{pk}^o)$$

(9)

Entonces la actualización de pesos es de la forma:

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \eta \delta_{pk}^o i_{pj}$$

(10)

### Momentum

Otra forma de incrementar la velocidad de aprendizaje es usando una técnica llamada *momentum*. Al calcular el valor del cambio de peso  $\Delta_p w$ , se añade una fracción del cambio anterior. Ese término adicional tiende a mantener los cambios de peso en la misma dirección (de ahí el nombre de *momentum*).

La ecuación de cambio de pesos de salida se modifican así:

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \eta \delta_{pk}^o i_{pj} + \alpha \Delta_p w_{kj}^o(t-1)$$

(11)

$\alpha$  : parámetro de momento (varía entre 0 y 1)

Los pesos en las capas ocultas se actualizan de acuerdo con las siguientes ecuaciones:

$$\delta_{pj}^h = f_j^{h'}(neta_{pj}^h) \sum_j \delta_{pk}^o w_{kj}^o$$

(12)

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

(13)

Puede observarse cómo una neurona modifica sus pesos de conexión para reducir el error, aun estando en una capa intermedia, donde no tiene contacto directo con la salida y, por tanto, con

el error producido. Esta es una de las grandes virtudes de las redes neuronales de propagación hacia atrás.

Un análisis similar al del *momentum* en la capa de salida, lleva a la siguiente ecuación para la oculta:

$$w_{ji}^h(t+1) = w_{ji}^h(t) + \eta \delta_{pj}^h x_{pi} + \alpha \Delta_p W_{ji}^h(t-1) \quad (14)$$

### **Término de error**

Como último paso se calcula el término de error para la red, el cual ya se ha usado para análisis y deducción de ecuaciones:

$$E_p = \frac{1}{2} \sum_{k=1}^M \delta_{pk}^2 \quad (6)$$

### **B. El caso de las nacientes**

El caso particular consiste en predecir caudales de nueve nacientes del manto acuífero herediano para determinar la producción total en un mes específico. Además se predecirá también el consumo y el número de abonados para cada mes. Los datos serán comparados (producción y consumo) para determinar si se dará subproducción, sobreproducción o si ambos datos coinciden; con base en ello se planificarán estrategias correspondientes en la Empresa de Servicios Públicos de Heredia (ESPH), como el uso de nuevos pozos en casos de subproducción, por ejemplo.

La necesidad de un método de predicción surge dado que las obras en acueductos, evidentemente, no pueden ser creadas de manera instantánea. La red neuronal preverá crecimientos o disminuciones en el consumo o en la producción acuífera; entonces, con la debida anticipación se podrán planificar y ejecutar las obras con el tiempo necesario, de modo que no se den situaciones imprevistas o de emergencia, en cuanto a la producción de las nacientes.

Se dispone de registros desde 1982 hasta 1996. Sin embargo los meses registrados no son los mismos para cada naciente o para los datos de consumo (véase el siguiente gráfico).

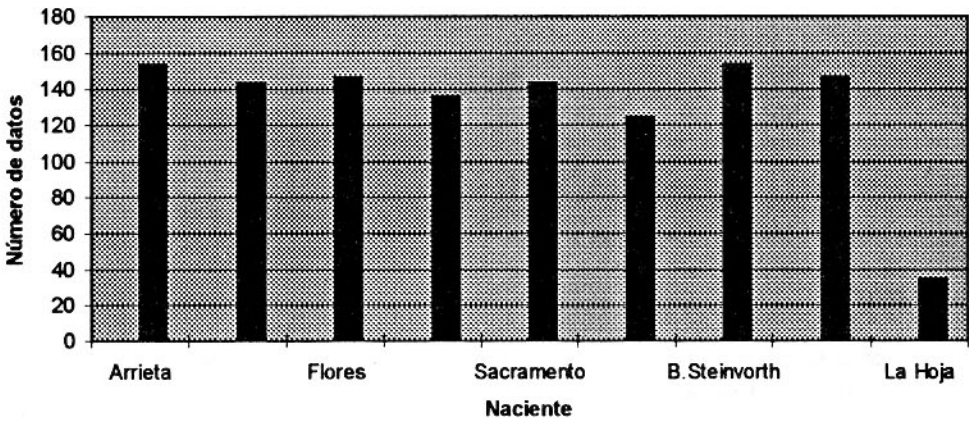


Gráfico 1 : Número de datos disponibles por naciente.

*¿Por qué no se utilizó para predicción, un promedio obtenido de la información de la base de datos históricos de la cual se dispone?*

El uso de un promedio o algún otro dato obtenido por fórmulas estadísticas no toma en cuenta la evolución de los caudales como lo hace una red neuronal. La red toma en cuenta el comportamiento por cada mes con respecto al mes anterior, al mes posterior y con respecto a los años contiguos; es decir se evalúa totalmente la evolución en el tiempo.

Aquí es importante recordar que para determinar qué clase de problemas se adaptan a los modelos conexionistas y redes neuronales debe pensarse cuáles problemas pueden ser

adecuadamente procesados por el cerebro humano. El caso de las nacientes se adaptaría bien a un aprendizaje humano (desde luego con un larguísimo período de aprendizaje), por lo cual se decide utilizar una red neuronal.

Además, los datos muestran un comportamiento totalmente no lineal año con año y mes por mes. Como ejemplo de esto se observa el comportamiento de la naciente Arrieta<sup>1</sup> a través de los años disponibles, en el siguiente gráfico. Nótese la dificultad de interpretar un comportamiento como éste mediante formulación matemática convencional con una precisión aceptable.

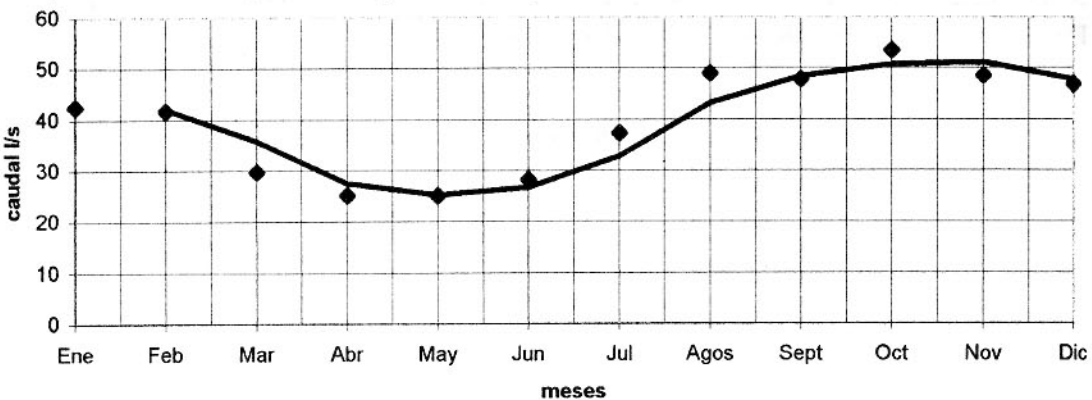


Gráfico 2: Producción promedio por mes de la naciente Arrieta



### ¿Por qué usar propagación hacia atrás?

Propagación hacia atrás resulta ser una técnica mediante la cual el comportamiento y la evolución de una serie de datos es "aprendida" y seguida por la red; es decir, la red simulará el comportamiento de cada nacimiento y el aumento en la población de abonados fielmente.

En contraste, otros métodos pueden hacer deducciones basados en métodos estadísticos que no tomarán en cuenta la evolución a través del tiempo. Además la red se fundamenta en procesamiento en paralelo, por lo cual supera en

eficiencia al procesamiento secuencial utilizado en los métodos convencionales de aproximación. El punto más importante en este aspecto es el hecho de que el comportamiento por modelar es totalmente no lineal, tal y como se observa en el gráfico N°2.

Es importante también destacar el hecho de que en una red de propagación hacia atrás todas las unidades de proceso intervienen en la disminución del error (no solo las de salida) por medio de la modificación de sus pesos, de ahí que el trabajo sea dividido en muchas pequeñas tareas que cada neurona realiza.

## 2. DISEÑO Y RESULTADOS ESPERADOS

### A. Diseño de la aplicación

La situación que se afronta en ESPH consiste en lo siguiente: se dispone de 9 nacientes para proporcionar agua a los abonados de Heredia y San Rafael, y se lleva un registro de la producción de las nacientes así como del número de abonados y consumo total de los mismos. Se desea tener un medio para predecir la producción de las nueve nacientes, así como del consumo y el número de abonados, con el fin de planificar estrategias de administración del acueducto.

Para la solución del problema se desarrollan 10 redes neuronales distintas, todas con 2 entradas

(mes y año) y 1 salida (producción de la nacimiento) o 2 (consumo).

Esto tiene como objetivo determinar si habrá subproducción, sobreproducción o producción justa en el mes observado; de este modo serán justificados proyectos de ampliación del acueducto y podrá realizarse una planificación anticipada y adecuada.

Todas las redes se manejan de modo similar de acuerdo con los siguientes parámetros:

- Número de capas ocultas: Se utiliza solo una capa oculta: inicialmente fue considerada la posibilidad de trabajar con más de una capa de unidades ocultas para el procesamiento de los datos, con el fin de aumentar la eficiencia de la red. Sin embargo se utilizó solamente una, fundamentado en que muchos autores han probado que una sola capa oculta es suficiente para aproximar cualquier función continua. Importa mucho, más que el número de capas, el número de neuronas en esta única capa oculta.
- Actualización de los pesos: La actualización de los pesos se hizo en línea, o sea, se actualizan los pesos después de evaluar cada patrón, a diferencia de la actualización *batch* (por lotes), donde se actualizan los pesos al finalizar la evaluación del último patrón. Esto se realizó así, ya que se toma en cuenta cada uno de los patrones y la influencia de cada peso en ellos.
- Registro del error: La observación del error es por lotes y no en línea; el error cuadrático se acumula durante todo un lote de patrones, luego se divide por el número de patrones (para obtener un promedio de los errores on-line en todo un lote) y se despliega y registra al final del último patrón evaluado. ¿Por qué? porque el registrar el error cuadrático por cada patrón produce una función de error de salida sumamente ruidosa (por la

oscilación del error de un patrón a otro dentro de un mismo lote).

- Número de unidades en la capa oculta: El número de neuronas ocultas se determinó de acuerdo con dos fórmulas heurísticas (la primera de Widrow y la segunda de Hinton), a saber:

$$H = \frac{0.1P}{E + S} \quad (15)$$

$$H = \frac{P \log_2(P)}{2(E + S)} \quad (16)$$

Donde:

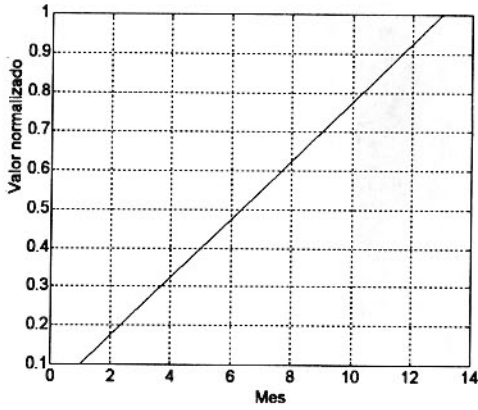
$H$  = Número de unidades en la capa oculta.  
 $P$  = Número de patrones de entrenamiento.  
 $E$  = Número de entradas de la red.  
 $S$  = Número de salidas de la red.

Ambas son promediadas para obtener el número de neuronas ocultas, tomando en consideración que de la primera se obtienen un número muy pequeño y de la segunda uno grande.

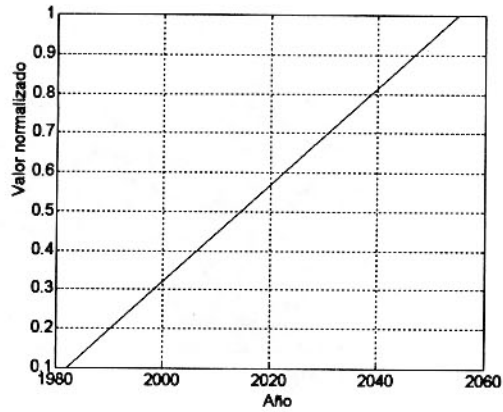
- Normalización: Para adaptar los datos a las entradas de las neuronas (deben estar entre 0 y 1) se utiliza una función de normalización lineal por cada entrada. En ella se asigna un valor de 0.1 al valor mínimo y un valor de 1.0 al máximo incrementado en una cantidad determinada de acuerdo con la situación. Así para la entrada de mes se asignó como máximo (es decir 1.0 normalizado) el valor de 13, el cual nunca se llega a evaluar; y para el año se utiliza como máximo 2055, debido a que la red predecirá hasta el año 2050, según requerimientos de la ESPH. Para las salidas (producción, consumo y número de abonados) se estimó un margen de aumento de un 30%, es decir, se asume un crecimiento futuro en las salidas de ese porcentaje como máximo, al cual se le asigna por lo tanto 1.0, mientras al valor mínimo de los patrones se asigna 0.1. Para comprender mejor lo anterior, refiérase a la siguiente tabla y los siguientes gráficos:

**Tabla 1 : Relación de máximos y mínimos con los valores normalizados.**

Valor normalizado	Año	Mes	Producción/Consumo/Abonados
0.1	1982	1	Mínimo
1.0	2055	13	Máximo * 1.3



(a)



(b)

**Gráfico 3 : Funciones de normalización para las entradas a la red: a) Meses b) Años**

• Selección del grupo de entrenamiento: No se utilizaron todos los datos disponibles para entrenar las redes; esto hubiera provocado un mayor número de unidades ocultas (véanse las fórmulas 15 y 16), con lo cual el número de cálculos por cada ciclo hubiera sido mucho mayor; además dadas las características de las redes de propagación hacia atrás será suficiente con un conjunto de patrones que represente el comportamiento de cada naciente. En el caso presente se seleccionaron los datos de

entrenamiento tomando en cuenta todos los meses y años, es decir, de cada año hay al menos cuatro datos, al igual que de cada mes. La distribución de los conjuntos de entrenamiento y validación de algunas de las redes se muestran en el gráfico N°4. Obsérvese que en el caso particular de La Hoja, el grupo de patrones disponibles se usó casi en su totalidad para entrenamiento, esto fue debido a que el número de patrones es relativamente pequeño (35 datos) comparado con los demás (alrededor de 150 datos cada uno).

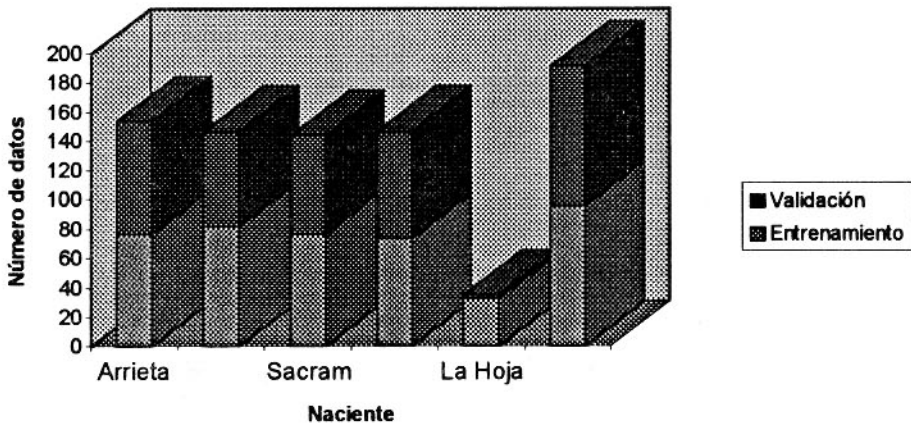


Gráfico 4 : Relación de grupos de entrenamiento y validación.

## B. Resultados esperados

Las expectativas de aprendizaje de las redes tomaron en cuenta los siguientes aspectos:

- Se utiliza un factor de aprendizaje inicial bastante pequeño (entiéndase entre 0 y 0.8). Conforme se va acercando al error cuadrático deseado,  $\eta$  disminuye para realizar un ajuste más fino de la red.
- Basado en el tipo de datos, el tamaño del grupo de entrenamiento y la capacidad de procesamiento (se utiliza una computadora con procesador Pentium de 133 MHz de velocidad con 24 MB de memoria RAM, sistema operativo Windows 95) se estima que se durará alrededor de un día con cada una de las redes para llegar a un error

aceptable (ver tabla N°2), basado en experiencias anteriores.

- No se utiliza el factor de *momentum*, en este caso, ya que no se sabe ciertamente cómo manipularlo para que ayude a la convergencia de la red.
- El error cuadrático al cual debe llegar cada red para un buen desempeño, se determina de acuerdo con la función de normalización de salidas de cada una y estimando un 1% de error con respecto al valor máximo de las salidas del grupo de entrenamiento, o con respecto al valor mínimo de las mismas. Idealmente se llegaría hasta un 1% de error con respecto al valor mínimo. A continuación, se muestra la relación de los errores mencionados en las redes más importantes diseñadas.

Tabla 2 : Valores de error cuadrático estimados.

Red	Valor Mínimo	Valor Máximo	Ep para errores de:	
			1% del Mínimo	1% del Máximo
La Hoja	70	425	8.52E-07	3.14E-05
Consumo	306555	760088	8.20E-06	5.04E-05
Flores	18	155	3.89E-07	2.89E-05

\*: Los valores máximo y mínimo de La Hoja y Flores están en litros por segundo (l/s) y los de consumo en metros cúbicos (m<sup>3</sup>).



En este punto cabe resaltar que, al pensar en generalización, debe tenerse claro qué se espera de una red que generalice bien.

Mucho entrenamiento puede causar sobreajuste, es decir se predicen muy bien los valores dentro del conjunto de entrenamiento, mas no así los valores provenientes de grupos externos al mismo (grupo de validación); lo cual no es conveniente para términos de extrapolación, como el que nos ocupa en este trabajo.

Una curva ajustada con muchos parámetros (pesos) sigue todos los detalles o el ruido, pero es pobre para interpolar o extrapolar.

### 3. RESULTADOS EXPERIMENTALES

El programa para generar las redes se implementó en *Visual Basic 4.0*, con todas las características descritas en los apartados

anteriores, de modo que permite manipular el aprendizaje y el *momentum* en línea mientras se está procesando la información y despliega el error cuadrático en pantalla al finalizar cada lote.

Mediante el mismo se montaron todas las redes de las nacientes y de consumo, con base en información teórica se manipularon el aprendizaje y el *momentum* de las mismas.

#### *Duración del entrenamiento*

La evolución del error, en las pruebas definitivas, se dio de acuerdo con lo esperado, según los parámetros utilizados; con excepción del tiempo de convergencia, que fue muchísimo mayor al estimado; así por ejemplo la red de predicción del caudal de la naciente Sacramento tuvo un comportamiento como el que se muestra a continuación:

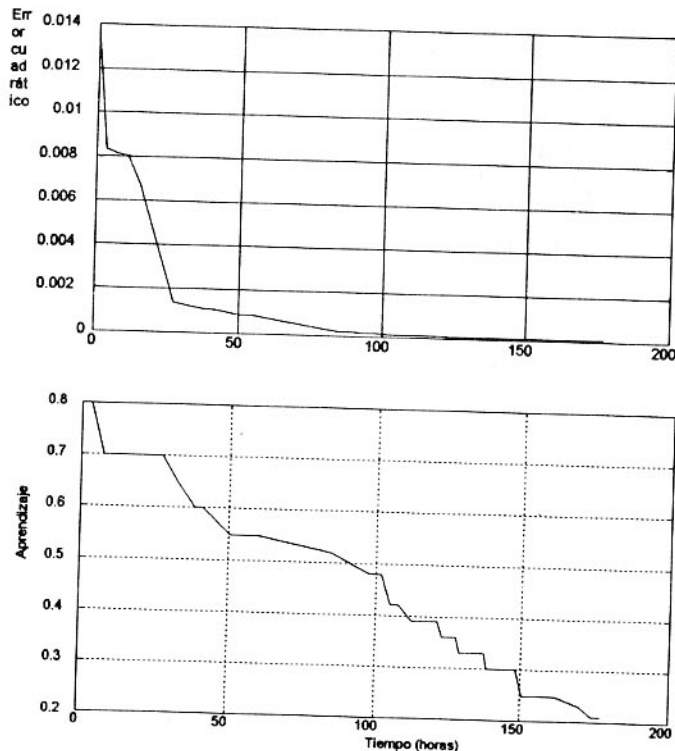


Gráfico 5 : Evolución del error (superior) y manejo del aprendizaje (inferior) para la red de Sacramento.

En este caso el factor de velocidad de aprendizaje inicial fue de 0.8, el cual fue disminuyendo conforme el error se acercaba a valores menores (esto con el fin de afinar la convergencia al error mínimo) hasta llegar a 0.15. Se observa cómo el aprendizaje disminuye con más frecuencia conforme el error se acerca más a cero, esto provoca un ajuste más fino en las proximidades del mínimo. El error cuadrático final al que se llegó fue de  $5 \times 10^{-4}$ . Ahora bien ¿por qué no se entrenó más la red para disminuir aún más el error?. El error alcanzado se considera suficientemente bueno para la predicción; además debe recordarse que al entrenar en exceso una red, esta evalúa muy bien los datos de entrenamiento, pero no generaliza bien, lo cual para términos de este proyecto no es adecuado del todo. Obsérvese también como en este caso el tiempo de entrenamiento fue de 175 horas (aproximadamente una semana de operación continua).

⇒ Es importante destacar aquí que, en redes neuronales, es frecuente el uso de términos como pequeño o grande (para los parámetros como factor de aprendizaje y *momentum*, y para los grupos de entrenamiento por ejemplo), suficiente o

adecuado (para el error). Estos términos no siempre corresponden a valores numéricos idénticos en cada caso particular; de hecho para cada grupo de entrenamiento la concepción de esos términos varía dependiendo del tamaño del grupo, las funciones de normalización, el propósito de la red, el tipo de datos y el comportamiento de los mismos dentro del grupo de entrenamiento; es decir, el tipo de función al que corresponden (como en las nacientes, donde es totalmente no lineal). Producto de esta experiencia puede incluirse dentro de esos términos palabras como “rápido” para la convergencia.

En este caso se considera un error (promedio con base en el error cuadrático) suficientemente pequeño para esta red, de  $5 \times 10^{-4}$  (observe que es mayor al error planificado con anterioridad) esto fue determinado probando datos del grupo de validación, de modo que el error fue adecuado (de 8.5% aproximadamente para todos los casos).

*Evaluación de las redes.*

A continuación se observan las evaluaciones de las redes con elementos del grupo de validación para la red de consumo.

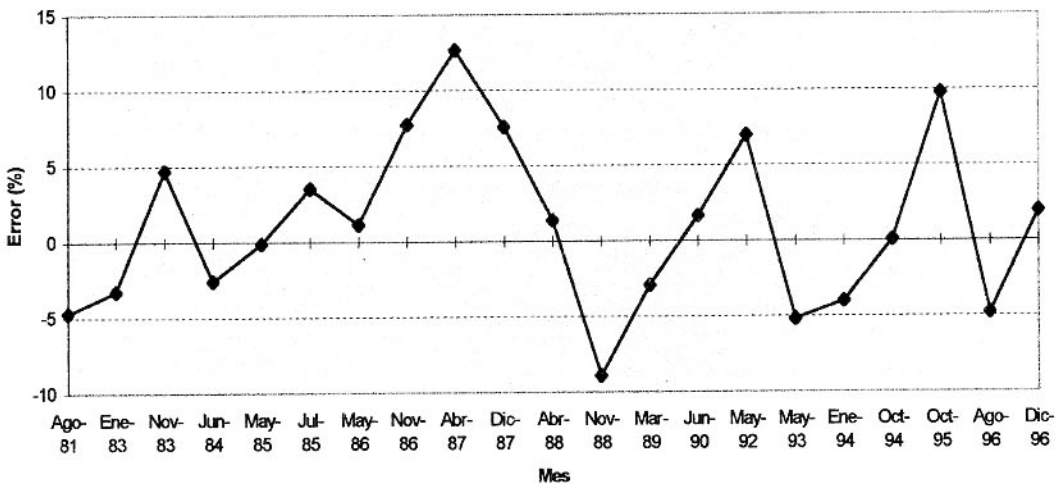


Gráfico 6: Porcentajes de error para muestras del grupo de validación (datos de Consumo).

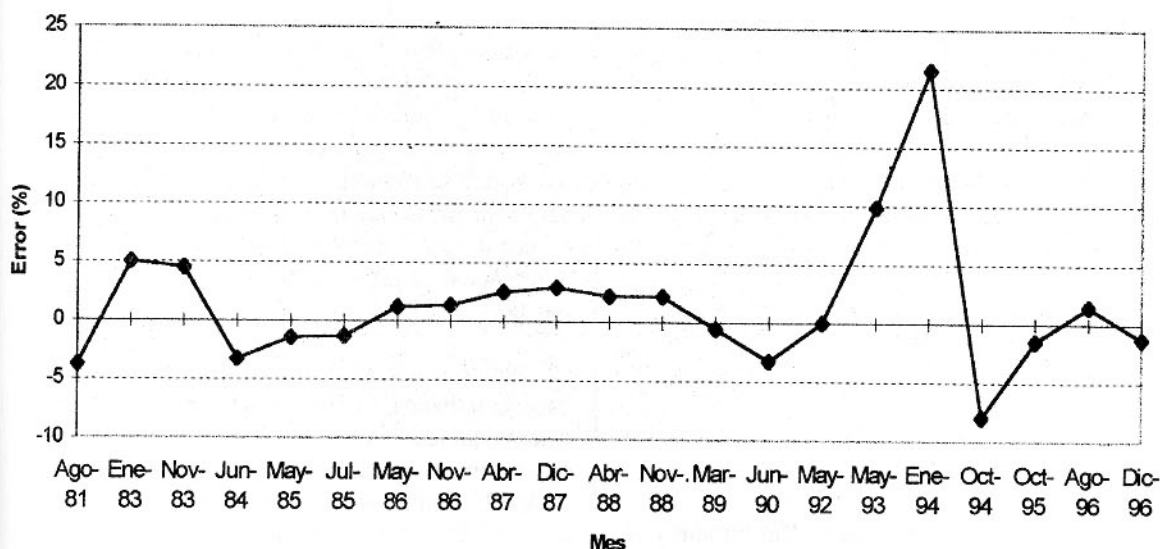


Gráfico 7: Porcentajes de error para muestras del grupo de validación (datos de Abonados).

Los promedios para estas muestras son 1.07 % y 1.44 % respectivamente, lo cual es muy satisfactorio. Con excepción de un dato en cada muestra, el resto de los errores no sobrepasa el 10 % en valor absoluto; en esta aplicación ese error se considera una buena aproximación al comportamiento del consumo y el número de abonados.

#### Sobreentrenamiento

Un aspecto importantísimo que se desprende de esta experiencia es el relativo a la calificación de sobreentrenada para una red neuronal. En el

caso de la naciente La Hoja, la red utilizada en el proyecto final presenta un error cuadrático mayor que otras sometidas a prueba también; sin embargo fue la elegida debido a que es la que mejor extrapola. Es decir, en las otras redes neuronales se dio un sobreentrenamiento, lo que provocó que no fueran útiles para predecir. ¿Cómo se comprobó esto? Se compararon los cuatro archivos de red (los cuales corresponden a distintos momentos del entrenamiento según la evolución del error) al evaluarlos con un dato del grupo de validación y los resultados fueron los siguientes:

Tabla 3: Evaluación de redes para la naciente La Hoja.

RED	ERROR CUADRÁTICO PROMEDIO	VALOR DE PREDICCIÓN (l/s)	ERROR RELATIVO (%)
Dato original (validación)	N.A.	202	0
Red 1	7.95 E-4	226.03	11.89
Red 2	5.07 E-4	271.42	34.36
Red 3	1.34 E-4	77.00	- 61.88
Red 4	1.06 E-4	70.4	- 65.14

N.A.: No Aplica.

No se utilizó el parámetro de *momentum*, debido a la naturaleza del proyecto: no se disponía de mucho tiempo para experimentar con un parámetro al cual la mayor parte de los investigadores y desarrolladores de redes neuronales se refieren solo teóricamente, ya que todas las experiencias se basan en la manipulación del factor de aprendizaje. Sin embargo se manipuló, y al hacerlo no varió significativamente el comportamiento durante el proceso de aprendizaje.

#### 4. CONCLUSIONES

Para aplicaciones a nivel comercial o industrial, se requiere de altísima capacidad de cómputo, con el fin de mejorar la velocidad de convergencia del error; lo cual llevará también a redes más precisas. Esto incluye desarrollar el programa en sistemas de 32 bits y en computadoras más rápidas y con mayor memoria de la utilizada.

Es recomendable, según lo experimentado en el desarrollo de este proyecto, iniciar siempre con factores de velocidad de aprendizaje no mayores a 1.0, y conforme el error disminuye, disminuir ese factor. Una opción factible para lograr ese propósito automáticamente (por programación) sería utilizar una función de decrecimiento exponencial, la cual se aplique al aprendizaje en línea. Sin embargo, la forma de aplicar la función exponencial dependerá del error deseado y por tanto se manejará distinto para cada caso particular.

El número de unidades de proceso en la capa oculta incide en la convergencia del error. Debe garantizarse una cantidad de neuronas suficiente para que el error disminuya adecuadamente. El método utilizado en este proyecto resultó ser indicado, por lo cual puede ser utilizado en experiencias posteriores.

La red neuronal es recomendable para casos que involucran datos a través del tiempo, debido a que toma en cuenta la evolución de los datos

dentro de todo el conjunto y no con respecto a un subconjunto, como en el caso de análisis estadísticos. Particularmente, la aplicación para predecir comportamientos propios de la naturaleza o de población resulta muy eficaz, por lo cual se recomienda la utilización de redes para solucionar este tipo de problemas, siempre y cuando se cuente con condiciones de información, similares a las estipuladas en este trabajo.

La selección del grupo de entrenamiento debe hacerse tomando en cuenta todo el intervalo de valores que cubren los patrones disponibles, de este modo se deducirá de manera más efectiva el comportamiento; es decir, no debe, por ejemplo, tomarse la primera mitad del conjunto de patrones para entrenamiento y el resto para validación.

De esta experiencia también se desprende que para propósitos de predicción, las redes de propagación hacia atrás se sobreentrenan fácilmente, de ahí que es recomendable que, con el fin de determinar hasta cuándo debe entrenarse, se pruebe la red con datos del grupo de validación hasta obtener una respuesta satisfactoria. A partir de ese punto se está propenso a un sobreentrenamiento, tal y como se comprobó aquí.

#### 5. FUTUROS TRABAJOS

Para expandir los horizontes de las redes neuronales en Costa Rica y utilizarlas más comercialmente, debe realizarse mayor investigación y experimentar en puntos tales como el manejo del *momentum*, el número de unidades ocultas que debe usarse, la selección del grupo de entrenamiento y el tiempo de entrenamiento, considerando la capacidad de cómputo con la cual se cuenta. Además es muy importante la identificación de los problemas que puedan resolverse por redes neuronales y la aplicación de métodos alternativos (dentro de las mismas redes) a propagación hacia atrás.



Tomando como punto de partida la experiencia de este proyecto, pueden implementarse más redes de propagación hacia atrás para predicción de caudales y de crecimientos de abonados en sistemas de acueductos locales o foráneos con la certeza de que los resultados serán satisfactorios.

Es indispensable también una mayor instrucción y capacitación a nivel práctico, con eventos como seminarios (orientados a la aplicación de propagación hacia atrás) que permitan intercambiar conocimientos con profesionales con mayor experiencia en el manejo de redes en el nivel comercial.

A nivel global para el diseño de redes, debe incrementarse el desarrollo de componentes y sistemas completos en equipo (*hardware*) y programas (*software*). Así como desarrollar más herramientas para el diseño de redes y para aplicaciones específicas como la descrita en este proyecto. Es de esperar que el desarrollo de redes dé lugar a un complemento para las computadoras tradicionales.

## 6. AGRADECIMIENTOS

Agradezco muy especialmente la oportunidad de realizar este proyecto a la empresa Automatización Industrial de Centroamérica S.A.; así como la valiosa ayuda en la elaboración y manipulación de las redes neuronales aquí presentadas.

## 7. BIBLIOGRAFÍA

Anderson, James y Rosenfeld, Edward; Neurocomputing: Foundations of Research; 1ª

Edición; Massachusetts Institute of Technology; Massachusetts; USA; 1988.

Bielawski, Larry y Lewand, Robert; Intelligent Systems Design; 1ª Edición; John Wiley & Sons; New York, USA; 1991.

Cruz, Claude; Understanding Neural Networks: A Primer; 2ª Edición; Cutter Information Corp.; Massachusetts, USA; 1991.

Freeman, James y Skapura, David; Redes Neuronales: Algoritmos, Aplicaciones y Técnicas de Programación; 1ª Edición; Addison Wesley; California, USA; 1991.

Hertz, John et al; Introduction of the Theory of Neural Computation; Primera Edición; Addison-Wesley Publishing Company; California; USA; 1991.

McClelland, James y Rumelhart, David; Explorations in Parallel Distributed Processing: a Handbook of Models, Programs and Exercises; MIT Press; Massachusetts; USA 1986.

McClelland, James y Rumelhart, David and the PDP Research Group; Parallel Distributed Processing: Explorations in the Microstructure of Cognition; Vol. II; MIT Press; Massachusetts; USA 1986.

Morgan, David y Scofield, Christopher; Neural Networks and Speech Processing, 1ª Edición; Kluwer Academic Publishers; Boston, USA; 1991.

Mozer, Michael; University of Colorado at Boulder, Entrevista Telefónica; 14 de agosto de 1997.