

Mapeo curvas típicas demanda de energía eléctrica del sector residencial, comercial e industrial de la ciudad de Medellín, usando redes neuronales artificiales y algoritmos de interpolación

Typical demand curvs of electric power for the residential, commercial and industrial sector of Medellin, using artificial neural networks and algorithms of interpolation

Héctor Tabares^{1}, Jesús Hernández²*

¹Departamento de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de Antioquia. Apartado Aéreo 1226.

²Universidad Nacional de Colombia, Calle 59A N°. 63-20. Apartado Aéreo 568. Medellín, Colombia.

(Recibido el 29 de abril de 2007. Aceptado el 30 de junio de 2008)

Resumen

Uno de los principales problemas para modelar el consumo de energía eléctrica en un lugar determinado, consiste en la extracción del conocimiento cuando éste se encuentra almacenado en grandes volúmenes de información como, por ejemplo, registros históricos. De acuerdo con esta representación, cada hecho ocurrido y registrado está compuesto por una pareja de componentes (t, P) en donde t representa el tiempo en el que se registro la muestra y P representa la potencia eléctrica consumida en ese instante. El registro diario cuenta con N casos que representa cada una de las parejas de estímulo-respuesta conocidas. El objetivo de este trabajo consiste en hallar una función que permita mapear el vector de variables de entrada t al vector de variables de salida P . donde F es una función cualquiera, en este caso el consumo de energía eléctrica. Su modelamiento con Redes Neuronales Artificiales (RNA) es un Perceptron Multi Capa (PMC). Otra forma de modelarlo es usando Algoritmos de Interpolación (AI).

----- **Palabras clave:** Redes neuronales artificiales, algoritmos de interpolación, demanda de energía eléctrica.

* Autor de correspondencia: Teléfono + 57 + 4 + 250 57 57, fax + 57 + 4 + 263 82 82, correo electrónico: htabares@udea.edu.co (H. Tabares).

Abstract

One of the main problems for modeling the electric power consumption in a certain place is the extraction of the knowledge when it is stored in big volumes of information like for example historical registrations. According with this representation, each fact happened and registered consists of a couple of components (t, P) where t represents the time of sample registration and P the electric power consumed at that time. The daily registration has N cases that each of the well-known stimulus-answer couples represents. The objective of this work is to develop a function that allows finding the vector of entrance variables t to the vector of exit variables P . F is any function, in this case the electric power consumption. Their modeling with Artificial Neural Network (ANN) is Multi a Perceptron Layer (PMC). Another form of modeling it is using Interpolation Algorithms(AI).

----- *Keywords:* Artificial neural networks, interpolation algorithms, electric power demand.

Introducción

Este artículo resuelve el problema de determinar las funciones que modelan la serie histórica de datos “Consumo de Energía Eléctrica, sectores Residencial, Comercial e Industrial de la ciudad de Medellín para un periodo de 24 horas”. La solución parte de la utilización de una RNA del tipo PMC como un aproximador universal de funciones y de la aplicación de los AI. Por lo tanto, este trabajo comienza haciendo una introducción a las RNA y a los AI. Seguidamente se realiza el planteamiento del problema a resolver y se estudian las soluciones propuestas usando RNA y AI. Finalmente se exponen las conclusiones.

Redes neuronales artificiales

La teoría y modelado de Redes Neuronales Artificiales esta inspirada en la estructura y funcionamiento de los sistemas nerviosos, donde la neurona es el elemento fundamental. Existen neuronas de diferentes formas, tamaños y longitudes, atributos importantes para determinar su función y utilidad [1, 2, 3].

Aprendizaje de una RNA

El aprendizaje es el proceso por el cual una Red Neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante el proceso de aprendizaje se reducen a la destrucción, modificación y creación de conexiones entre las neuronas. En el caso de las Redes Neuronales Artificiales, se puede considerar que el conocimiento se encuentra representado en los pesos de las conexiones. En realidad puede decirse que se aprende modificando los valores de los pesos de la red. Durante el proceso de aprendizaje, los pesos de las conexiones de la red sufren modificaciones, por tanto se puede afirmar que este proceso ha terminado (la red ha aprendido) cuando los valores de los pesos permanecen estables. En forma general, se consideran dos tipos de aprendizaje: Supervisado y no supervisado. La diferencia fundamental entre ambos tipos estriba en la existencia, o no, de un agente externo (supervisor) que controle el proceso de aprendizaje de la red. Par-

ticularmente las supervisadas se caracterizan por tener arquitecturas en niveles y conexiones entre las neuronas estrictamente hacia delante. Frecuentemente son utilizadas para clasificar patrones.

Algoritmos de aprendizaje: Existen dos categorías de algoritmos de aprendizaje: el Descenso por gradiente, y técnicas de entrenamiento de segundo orden (Gradiente Conjugado, Levenberg-Marquardt -LM).

Todos los algoritmos parten de una condición inicial x_0 y luego se modifican por etapas de acuerdo con:

$$x_{k+1} = x_k + \alpha p_k$$

donde p_k representa una dirección de búsqueda y el escalar positivo α es la tasa de aprendizaje, que determina la longitud del paso. El entrenamiento finaliza cuando una de las siguientes condiciones se presenta:

- 1 Cuando se ha alcanzado la cantidad máxima de iteraciones.
- 2 Cuando se ha alcanzado el error final deseado.
- 3 La ejecución del gradiente alcanza un valor por debajo del mínimo gradiente.

Ventajas y limitaciones de las RNA

El éxito de las RNA se debe a que pueden modelar datos que:

- Exhiben alta no linealidad impronosticable.
- Son caóticos en el sentido matemático.

Uno de los problemas que presenta el algoritmo de entrenamiento regla delta generalizada para redes multicapa es que busca minimizar la función de error, pudiendo caer en un mínimo local o en algún punto estacionario, con lo cual no se llega a encontrar el mínimo global de la función del error. Sin embargo, se debe tener en cuenta que no tiene por qué alcanzarse el mínimo global en todas las aplicaciones, sino que puede ser suficiente con un mínimo error preestablecido. El algoritmo basado en LM es el más rápido para redes backpropagation cuando se trabaja con un moderado número de parámetros de la red. Por otra parte, requiere mayor costo computacional debido a que implica

el cálculo de matrices inversas. Tiene la desventaja de requerir de unas tuplas de entrenamiento lo más estándar posible, pues de otra forma sólo aproximará correctamente valores que se encuentren dentro de los patrones de aprendizaje. La utilización de modelos de aproximación con RNA, puede ocurrir el sobre ajuste (overfitting), que es un problema de los modelos estadísticos. *Esto es una mala situación porque en lugar de aprender a aproximar la función presente en los datos, la función simplemente los memoriza. El ruido en las series históricas se aprende entonces como parte de la función, a menudo destruyendo su habilidad para generalizar.*

Es útil mencionar que las comparaciones de velocidad entre diferentes algoritmos de entrenamiento no siempre son claros, ya que diversos autores han usado, para presentar sus resultados, diferentes: problemas, criterios al considerar la red entrenada, medidas de velocidad computacional, y aproximaciones o concepciones, al promediar todos estos resultados.

Algoritmos de interpolación

La interpolación se usa para obtener datos intermedios a partir de una tabla en la cual los valores del conjunto de llegada (Y_i) son conocidos con gran precisión. Las funciones que suelen utilizarse para interpolar se seleccionan, principalmente, como funciones polinómicas [4, 5]. En caso de que la formulación analítica del problema sea conocida, se trata de obtener los coeficientes óptimos que permitan ajustar dicha formulación al conjunto de datos.

El polinomio de interpolación de grado n se podrá escribir en forma genérica como:

$$p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n = \sum_{i=0}^n a_i x^i$$

Ventajas y limitaciones de las AI

Una dificultad práctica que ocurre con la interpolación consiste en que el término de error de la aproximación es difícil de aplicar. Generalmente el grado del polinomio necesario para lograr la exactitud deseada no se conoce.

Planteamiento del problema

La calidad de servicio de un sistema eléctrico [6] puede cuantificarse a través de varios parámetros relacionados con: la continuidad del servicio, las fluctuaciones de voltajes, el contenido armónico de las formas de onda de voltaje y de corriente, variaciones de frecuencia, y la regulación. El concepto de calidad del servicio es bastante amplio, de manera que no es posible sintetizarlo en un solo parámetro o índice. Un parámetro comúnmente utilizado para evaluar la calidad del servicio, consiste en monitorear periódicamente los registros de consumo de energía eléctrica. El estudio de su comportamiento sirve para tomar medidas que garanticen el rendimiento óptimo del sistema en su conjunto. Para el caso de la ciudad de Medellín las figuras 1, 2 y 3 muestran el consumo de energía eléctrica para los sectores residencial, comercial e industrial en un período de 24 horas.

Como se observa en las figuras 1, 2 y 3, la dificultad radica en la extracción de la información pertinente cuando ésta se encuentra almacenada en grandes volúmenes, como por ejemplo, registros históricos gráficos o tabulados. La confiabilidad del análisis aumentaría si se tuviera una función con la cual modelar cada serie histórica.

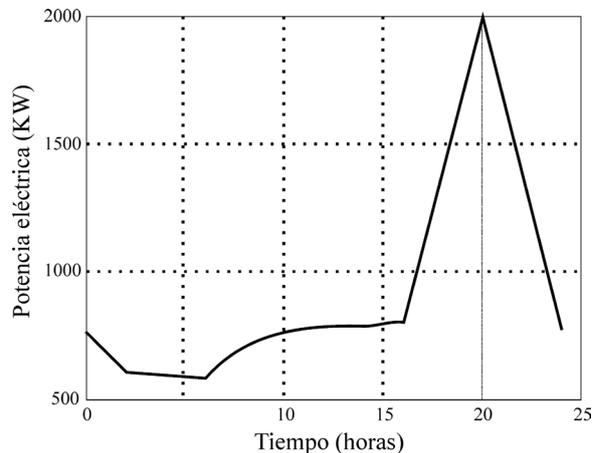


Figura 1 Demanda diaria de energía eléctrica, sector residencial

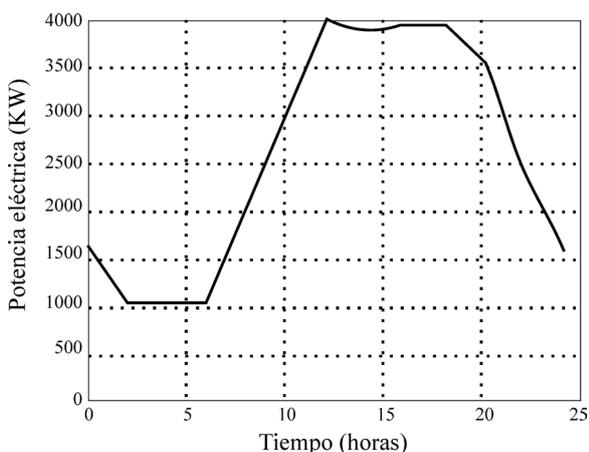


Figura 2 Demanda diaria de energía eléctrica, sector comercial

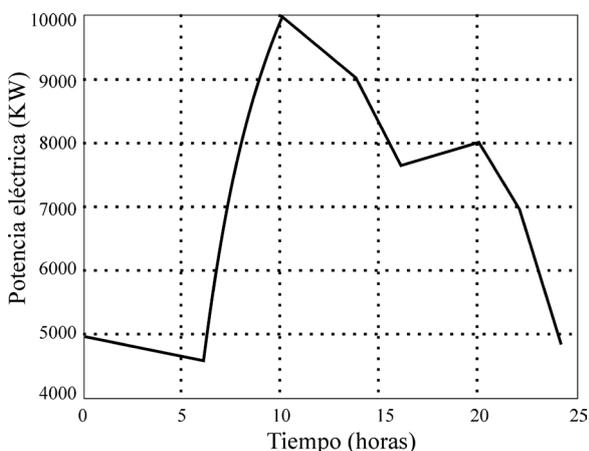


Figura 3 Demanda diaria de energía eléctrica, sector industrial

Aproximación a la serie histórica de datos consumo de energía eléctrica en la ciudad de Medellín

En este trabajo investigativo se encontró que la mejor solución para mapear series de datos son las RNA del tipo PMC y los AI. Su utilización significó aclarar las principales limitaciones que presentan ambos métodos. Con respecto a las RNA se tiene que a pesar de que la investigación desarrollada en los últimos años sobre redes neuronales ha llevado al descubrimiento de

varios resultados teóricos y empíricos significativos, el diseño de las arquitecturas o topologías de las RNA para aplicaciones específicas bajo un conjunto dado de restricciones de diseño es un proceso de prueba y error dependiendo principalmente de la experiencia previa con aplicaciones similares [7].

Por otra parte, el desempeño de una red neuronal sobre problemas particulares es críticamente dependiente, entre otras cosas, del número de ejemplos de entrenamiento, la complejidad de la función a ser resuelta, el valor inicial de los pesos, el valor del coeficiente de aprendizaje y del algoritmo de aprendizaje utilizado.

Sabido lo anterior, se presentan a continuación las condiciones iniciales empleadas para aproximar las curvas típicas de demanda de energía con RNA. Para realizar los experimentos numéricos de aproximación se utilizó el paquete informático MATLAB. Éste es un conjunto de programas matemáticos y se pueden realizar programas usando lenguaje de alto nivel. Incluye una completa librería de funciones para trabajar con RNA (*Toolbox Neural Network, versión 6.5*), lo que lo convierte en una herramienta informática ideal para los requerimiento planteados en este trabajo investigativo.

Función de activación

La función de activación en cada neurona es la función sigmoideal, como se ilustra en la figura 4.

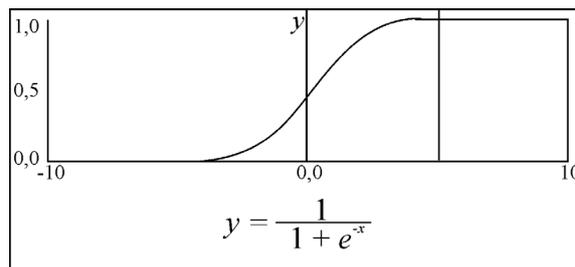


Figura 4 Función sigmoideal

Por tanto, es necesario escalar las variables de entrada y de salida, como se ilustra a continuación.

Escalando vector de entradas y salidas

$$x_e = \frac{2 * LS_EjeX(x - ValorMinVectorX)}{ValorMaxVectorX - ValorMinVectorX} + LI_EjeX$$

$$y_e = \frac{LS_EjeY(y - ValorMinVectorY)}{ValorMaxVectorY - ValorMinVectorY} + LI_EjeY$$

Definiendo

LS: Límite_Superior

LI: Límite_Inferior

Inicializando las variables:

LS_EjeX= +0.95.

LI_EjeX=-0.95.

LS_EjeY= 0.9.

LI_EjeY=- 0.05

Topología y Algoritmos de entrenamiento

Para resolver los problemas de aproximación expuestos en las figuras 1, 2 y 3, se utilizará un criterio heurístico, basado en la intuición y la experimentación, para seleccionar la topología de la

```
%UNIVERSIDAD DE ANTIOQUIA
%FACULTAD DE INGENIERÍA
%Programa Ingeniería Eléctrica
%Asignatura : Control Inteligente
```

```
%-----
```

```
%Variable de entrada (tiempo) escalada
```

```
P=[-0.9 -0.7 -0.6 -0.4 -0.3 -0.1 0.0 0.1 0.3 0.4 0.6 0.7 0.9]
```

```
%Variables de salida escalada (Potencia consumida en KW sector Residencial, Comercial e Industrial)
```

```
T=[ 0.1 0.2 0.0; 0.0 0.0 0.0; 0.0 0.0 0.0; 0.0 0.0 0.0; 0.1 0.3 0.6; 0.1 0.8 0.8;
    0.2 0.9 0.7; 0.2 0.9 0.6; 0.2 0.8 0.4; 0.5 0.9 0.5; 0.9 0.8 0.5; 0.7 0.4 0.3; 0.2 0.2 0.0]
```

```
plot(P,T,P,T,':')
```

```
PR=[min(P) max(P)];
```

```
%definir arquitectura de la red
```

```
net = newff(PR,[5 3],{'logsig' 'logsig'},'trainlm');
```

RNA. Ésta tendrá una neurona de entrada (tiempo), tres de salida (Potencia en Kw del sector Residencial, Comercial e Industrial) y una capa oculta con 5 neuronas. Las neuronas adaptativas son opcionales, por lo cual las simulaciones se harán sin estas conexiones, como se ilustra en la figura 5.

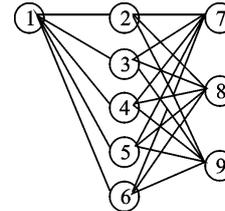


Figura 5 Topología RNA

En todas las pruebas de validación realizadas, los pesos del PMC se inicializarán aleatoriamente una sola vez. Se empleará el algoritmo de entrenamiento de segundo orden *Trainlm* por ser considerado el más rápido, con máximo 3000 iteraciones y un error final en la aproximación de 0.001.

Código fuente MATLAB

El código fuente en MATLAB, para simular la aproximación a las curvas estudiadas en este artículo, usando RNA del tipo PMC es como se presenta a continuación.

%para ver parámetros de entrenamiento

```
net.trainParam.epochs = 3000;
net.trainParam.goal = 0.001;
net.trainParam.min_grad = 1e-9;
net.trainParam.show=50;
```

%Entrenando la red

```
net = init(net)
net = train(net,P,T);
```

%Datos de Generalización

```
ValorMin = 0; ValorMax = 24; Valor = 0; i = 1;
```

```
while Valor <=ValorMax
```

%Escala la entrada

```
G(i) = ((1.9*(Valor - ValorMin))/(ValorMax - ValorMin))-0.95;
i = i + 1; Valor = Valor + 0.01;
```

```
end
```

% definir patrones prueba de generalización

```
PG = G(1,:);
```

%Validando el entrenamiento

```
Y = sim(net,PG);
plot(P,T,PG,Y, 'r')
```

Gráficas de aproximación usando RNA

La figura 6 muestra las curvas de aproximación a las series históricas consumo de energía eléctrica de los sectores Residencial, Comercial e Industrial de la ciudad de Medellín, modeladas con RNA del tipo PMC.

Una de las características de las RNA es su capacidad de generalización, es decir la facultad de la red para responder apropiadamente cuando se le presentan datos o situaciones a los que no había sido expuesta anteriormente. El sistema puede generalizar la entrada para obtener una respuesta. Esta característica es muy importante cuando se tiene que solucionar problemas en los cuales la información de entrada es poco clara; además permite que el sistema de una solución incluso cuando la información de entrada esta especificada de forma incompleta. Como se observa en la figura 6, la RNA mapeo, generalizó, correctamente el universo de datos de entrada. Nótese que el error final en las aproximaciones fue de 0,01, lo cual es considerado como una buena medida.

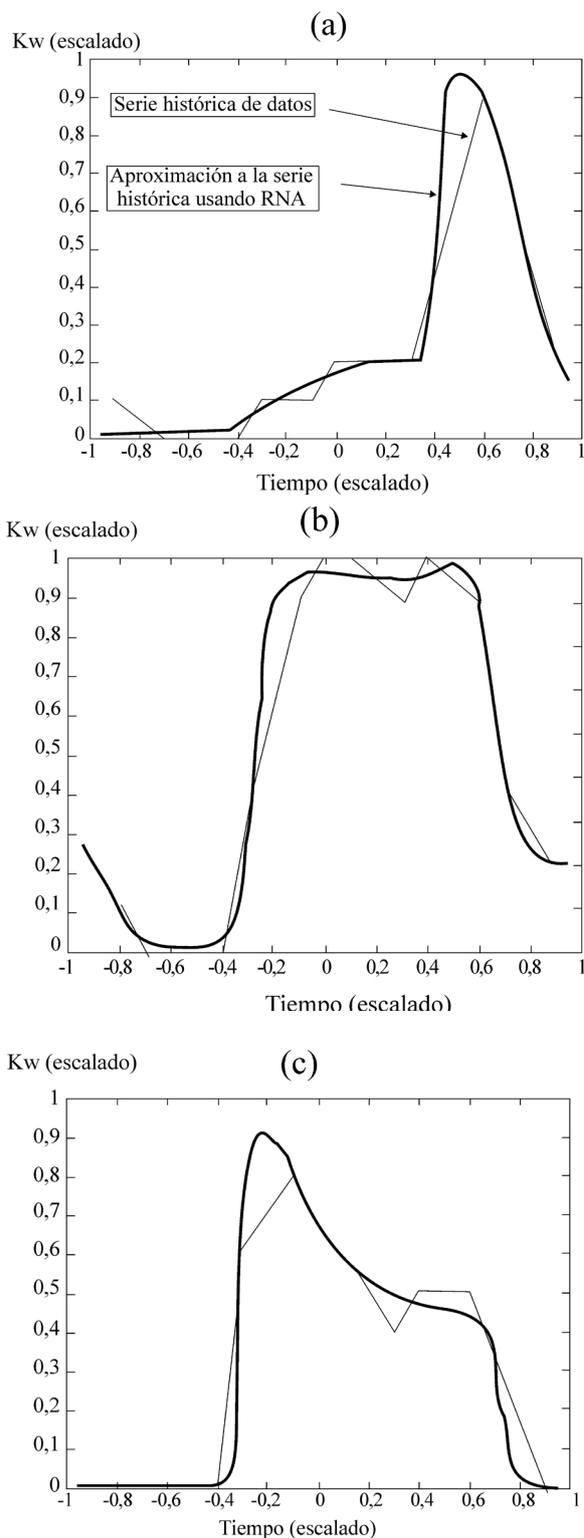


Figura 6 Salida de la RNA, (a) sector Residencial, (b) Comercial, (c) Industrial

Algoritmos de Interpolación

Con el objeto de validar las series de datos estudiadas en este artículo usando AI [6, 7], realizar experimentos numéricos y pruebas de ensayo, se desarrolló el simulador software sobre Métodos Numéricos MN_UdeA.exe. Éste incluye el módulo AI. Con el objeto de realizar una segunda validación de los datos obtenidos en este artículo usando LD, realizar experimentos numéricos y pruebas de ensayo, se desarrolló el simulador software sobre Lógica Difusa usando el lenguaje de programación VC++ y titulado LD_UdeA.exe. La implementación software se encuentra en la dirección electrónica [8] de la Universidad de Antioquia:

La interfaz del sistema es como se ilustra en la figura 7.

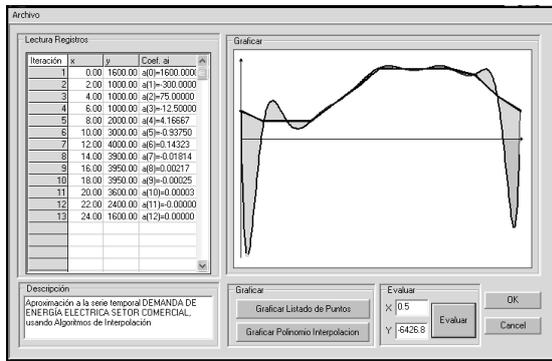


Figura 7 Interfaz programa MN_UdeA

En la figura 8, se ilustra la aproximación a las curvas típicas de consumo de energía sector residencial, comercial e industrial de la ciudad de Medellín, usando AI. Como se observa en la figura 8, las aproximaciones obtenidas usando Algoritmos de Interpolación, adolecen de la capacidad para generalizar correctamente el universo de datos. Nótese que el error final en la aproximación no se pudo calcular, pues para lograr la misma exactitud de 0.01 obtenida con RNA, el grado del polinomio es desconocido. Por lo tanto lo que se acostumbra es obtener los resultados de varios polinomios, hasta que se logre una correspondencia adecuada con el ejemplo a aproximar. Además el trabajo realizado al calcular la aproximación mediante

un segundo polinomio no reduce el que se requiere para calcular el tercero; tampoco es más fácil obtener la cuarta aproximación, una vez conocida la tercera y así sucesivamente.

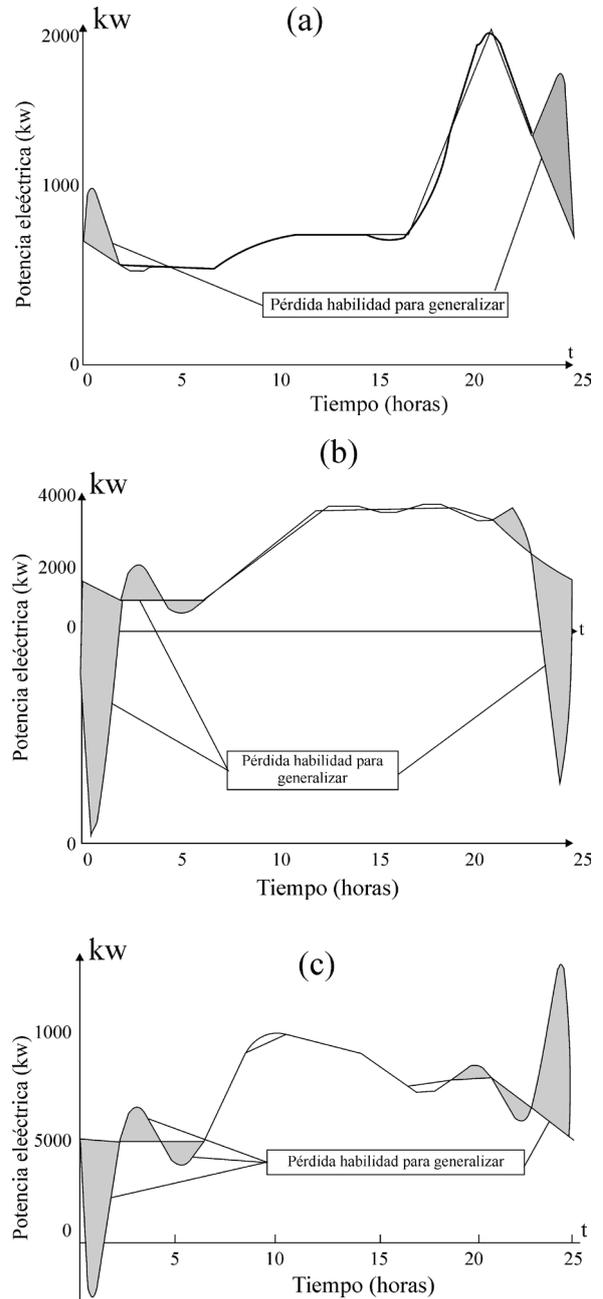


Figura 8 Aproximación a las curvas (a) sector residencial, (b) comercial (c) industrial usando algoritmos de interpolación

Conclusiones

El concepto de calidad de servicio de un sistema eléctrico esta adquiriendo especial importancia en la ciudad de Medellín, dada la presencia de una mayor cantidad de cargas sensibles tanto en las variaciones de voltaje y frecuencia como a los cortes de suministro. Existen diversos cuantificadores que dan cuenta de las fluctuaciones lentas o rápidas de voltaje e indican la necesidad de tomar medidas correctivas, dado que las fuentes de estos problemas son conocidas.

Por otra parte, los cortes de suministro de energía eléctrica en los sectores residencial, comercial y/o industrial afectan a los usuarios produciendo graves distorsiones en el desarrollo habitual de cualquier actividad. Éstas se cuantifican midiendo la frecuencia de aparición de cortes de suministro y su duración.

Con el objeto de evaluar el impacto de una falla en el sistema eléctrico, el operador del sistema estudia los registros históricos de consumo. El principal problema consiste en la extracción de la información pertinente cuando se encuentra almacenada en grandes volúmenes, siendo fuente de posibles errores en su lectura, que finalmente no se detectan.

Como una forma de contribuir a la confiabilidad del estudio sobre los registros de consumo eléctrico, se propone en este artículo mapear las curvas típicas de consumos de energía eléctrica en los sectores residencial, comercial e Industrial en la ciudad de Medellín usando RNA del tipo PMC. Su utilización, ayuda al analista del sistema eléctrico a evaluar, de manera más ágil, la variación de consumo de carga eléctrica de los diferentes sectores y el impacto que tendría en éstos una falla.

Como quedó demostrado en las pruebas de aproximación usando AI, éstos no se recomien-

dan, debido a que no se pudo calcular los errores al generalizar tendencias.

Finalmente, con respecto a las RNA es necesario comentar que son sistemas caóticos donde todo influye en todo y todo esta interconectado con todo. Por tal razón, las investigaciones que se realizan a nivel mundial tendientes a mejorar su desempeño consisten en determinar el valor inicial óptimo de los pesos de la red, el valor inicial óptimo del valor del coeficiente de aprendizaje, el mejor conjunto de datos de entrenamiento de la red, los mejores algoritmos de entrenamiento, la habilidad que tiene la red para aprender y generalizar, la topología de una RNA con la cual resolver un problema particular.

Referencias

1. J. Hilara. "Redes Neuronales Artificiales". *Fundamentos, modelos y aplicaciones*. Ed. Alfa Omega. Madrid. 2000. pp. 132-153.
2. B. Martín del Brio. *Redes Neuronales y Sistemas Difusos*. Ed. Alfa Omega. Madrid. 2002. pp 64 - 69.
3. T. Masters. *Practical Neural Networks recipes in C++*. Ed. Academic Press, Inc. San Diego (CA). 1993. pp. 173-180.
4. R. Burden, F. Douglas. *Análisis Numérico*. Ed. Thomson Learning. México. 2002. pp. 104-141.
5. J. Mathews, K. Fink. *Métodos Numéricos con Matlab*. 3ª ed. Ed. Prentice Hall. Madrid. 2000. pp. 203-250.
6. S. Grainger. *Análisis de sistemas de potencia*. Ed. M. Graw Hill. New York. 2002. pp. 56-124.
7. R. Dow, J. Sietsman. "Creating Artificial Networks that generalize". *Neural Networks*. Vol. 4. pp. 198-209.
8. Facultad de Ingeniería, Universidad de Antioquia. <http://jaibana.udea.edu.co/producciones/programas.html>. Consultada el 4 de Marzo de 2007.