

Pronóstico de Carga Eléctrica usando Redes Neuronales Artificiales
Robert Gerardo Castro Salguero

Sección de Postgrado
Facultad de Ingeniería Eléctrica y Electrónica
Universidad Nacional de Ingeniería
Lima, Perú

Resumen

Las Redes Neuronales Artificiales, han sido aplicado exitosamente en diversas áreas incluyendo reconocimiento de habla y de patrones, ingeniería eléctrica, robótica y pronósticos de mercado. La Red Neuronal más usual es aquella que obtiene conocimiento de la experiencia. La experiencia es proporcionada a la red en forma de datos de entrenamiento. Una vez entrenada, la red Neuronal puede reconocer datos que no ha visto antes. Este trabajo presenta una introducción a la manera en la cual trabaja la red Neuronal y como usarla en el pronóstico de carga.

La RNA es usada para encontrar una relación entre las cargas y temperaturas. por lo que requiere datos para su entrenamiento. En la actualidad la RNA proporciona mejor performance que otras técnicas de pronóstico.

1. Introducción

Diversas técnicas para predicción de carga de sistemas de potencia han sido propuestos en las últimas décadas. La Predicción de carga de unos minutos a varios días, ayuda al operador del sistema para una eficiente programación. Además, el pronóstico provee información para ser usada en posibles intercambios de energía con otros sistemas. Además, el pronóstico de la carga es usado para la seguridad del sistema, lo cual puede proporcionar valiosa información para detectar muchas situaciones vulnerables de antemano.

Tradicionales aproximaciones, tales como regresión e interpolación, pueden no dar resultados precisos. Por otro lado, algoritmos complejos con pesada carga computacional pueden converger lentamente y divergir en ciertos casos.

Varios algoritmos han sido sugeridos para el pronóstico de carga. Estos pueden ser clasificados en dos categorías de acuerdo a las técnicas. Una aproximación trata el patrón de carga como una señal de serie de tiempo y predice la futura carga usando varias técnicas de análisis de series de tiempo. La segunda aproximación reconoce

que el patrón de carga es fuertemente dependiente de las variables del clima, y encuentra una relación funcional entre las variables del clima y la carga del sistema. La futura carga es entonces pronosticada insertando información pronosticada de clima en la predeterminada relación funcional.

En la aproximación mediante series de tiempo se produce una imprecisión de la predicción e inestabilidad numérica. Una de las razones de la imprecisión es que no utiliza información del clima. Hay una fuerte correlación entre la conducta del consumo de potencia y las variables climáticas tales como temperatura, humedad, velocidad del viento y las nubosidad. Esto es especialmente cierto en áreas residenciales. Las aproximaciones mediante series de tiempo mayormente utilizan algoritmos adaptivos orientados a matrices, computacionalmente pesados, que en ciertos casos pueden ser inestables.

Muchas aproximaciones de regresión intentan encontrar una relación funcional entre las variables climáticas y la demanda de carga. Esta técnica usa una representación lineal o lineal por tramos para la función de predicción. Para una combinación lineal de estas representaciones, la aproximación por regresión busca la relación funcional entre las variables de clima seleccionadas y la demanda de carga. Técnicas convencionales asumen, con justificación una relación lineal. La relación funcional entre la carga y variables climáticas, sin embargo, es no estacionario, pero depende de los elementos espacio-temporales. La aproximación convencional por regresión no tiene la versatilidad para direccionar su variación temporal. Es decir, mas bien producirá un resultado promedio. Por lo tanto, una técnica adaptiva será necesaria.

Aquí se presenta un algoritmo que utiliza una RNA, con perceptron y una capa. La RNA traza un patrón de carga y predice (extrapola) un patrón de carga usando recientes datos de carga. Este algoritmo usa una aproximación del clima para el modelamiento de la información. La RNA es apta para modelamiento no lineal y

adaptivo. Esto no requiere asumir alguna relación funcional entre la carga y las variables climáticas de antemano. Podemos adaptar la RNA para exponerla a nuevos datos. La RNA esta siendo utilizada como herramienta en otros problemas de sistemas de potencia tal como la seguridad, identificación de armónicos, procesamiento de alarmas, diagnóstico de fallas, etc.

Se hará una breve revisión de los diversos algoritmos de predicción de carga. La regla delta generalizada usada para entrenar la red es mostrada mas adelante.

Luego definiremos el problema de predicción de carga, mostraremos la topología de la RNA usada en la simulaciones y analizaremos la performance en términos de errores (la diferencia entre el valor actual y el valor pronosticado). Mas adelante presentaremos nuestros resultados y conclusiones.

2. Técnicas para el Pronóstico de la Carga Eléctrica

2.1 Series de tiempo.-

La idea de la aproximación por series de tiempo esta basada en el hecho que un patrón de carga no es nada más que una señal en series de tiempo con periodicidad estacional, semanal y diaria. Esta periodicidad da una grosera predicción de la carga. La diferencia entre la carga actual y pronosticada carga puede ser considerada como un proceso estocástico. Las técnicas usadas para el análisis de esta señal aleatoria incluye:

- ❖ El filtro de Kalman
- ❖ El método Box-Jenkins
- ❖ El movimiento auto-regresivo promedio (ARMA)
- ❖ Técnicas de expansión espectral.

La aproximación de filtro de Kalman requiere estimación de una matriz de covarianza. La posible alta no estacionalidad de los patrones de carga, sin embargo, puede impedir una adecuada estimación.

El método Box-Jenkins requiere la función de autocorrelación para identificar propios modelos ARMA. Esto puede ser realizado usando técnicas de reconocimiento de patrones. El mayor obstáculo es aquí la baja performance.

El modelo ARMA es usado para describir el comportamiento estocástico del patrón de carga horaria en un sistema de potencia. El modelo ARMA asume que la carga por

hora puede ser estimado por una combinación lineal de las ultimas horas. Generalmente, a más datos, mejor es el resultado en términos de precisión. Mucho tiempo computacional es requerido para la identificación de parámetros.

Las técnicas de expansión espectral utilizan las series de Fourier. Desde que el patrón de carga puede ser considerado aproximadamente como una señal periódica, puede ser descompuesto en cierto numero de sinusoides con diferentes frecuencias. Cada senoide con una especifica frecuencia representa una base ortogonal. Una combinación lineal de estas bases ortogonales con sus propios coeficientes puede representar perfectamente un patrón de carga. Sin embargo los patrones de carga no son perfectamente periódicos. Esta técnica usualmente emplea solamente una pequeña fracción del posible conjunto de la base ortogonal, y por lo tanto es limitado a señales de poca variación. Abruptos cambios en el clima causan rápidas variaciones del patrón de carga el cual resulta en componentes de alta frecuencia en dominio de la frecuencia. Por lo tanto, la técnica de expansión espectral puede no proveer una adecuada predicción para el caso de rápidos cambios de clima cuando no son usados suficientes elementos base. Generalmente, técnicas en series de tiempo hacen buen trabajo a menos que, se de un abrupto cambio en las variables ambientales y sociales, las cuales afectaran el patrón de carga. Si hay algún cambio en estas variables, la técnica de series de tiempo no es muy usual. Por otro lado, estas técnicas usan un gran numero de complejas relaciones, requiere un gran tiempo computacional y pueden producir inestabilidad numérica.

2.2 Regresión

El procedimiento general para la aproximación por regresión es:

- 1) Seleccionar las variables disponibles
- 2) Asumir elementos funcionales básicos
- 3) Encontrar coeficientes para la combinación lineal

Desde que la temperatura es la información más importante de todas las variables climáticas, ésta es más usada comúnmente en la aproximación por regresión (posiblemente no lineal). Sin embargo, si usamos variables adicionales tales como humedad, velocidad del viento, y

nubosidad, mejores resultados deberán ser obtenidos.

Muchas aproximaciones por regresión usan aproximación lineal o lineal por tramos como elementos funcionales básicos.

Una relación funcional usualmente empleada entre la carga, L y la temperatura, T , es:

$$L = \sum_{i=1}^N a_i T \{U(T - T_{i1}) - U(T - T_{i2})\} + C$$

Donde :

$$U(T) = \begin{cases} 1, & \text{si } T \geq 0 \\ 0, & \text{en otros casos} \end{cases}$$

y a_i, T_{i1}, T_{i2} , y C son constantes,

y $T_{i1} > T_{i2}$ para todo i

Las variables (L , a_i , T_{i1} , T_{i2} , y C) están variando temporalmente. La dependencia de tiempo, sin embargo, no se nota explícita por razones de la notación compacta.

Después que la forma funcional básica de cada subclase de rango de temperatura es decidido, los propios coeficientes de la forma funcional son encontrados a fin de hacer una combinación lineal representativa de las funciones base.

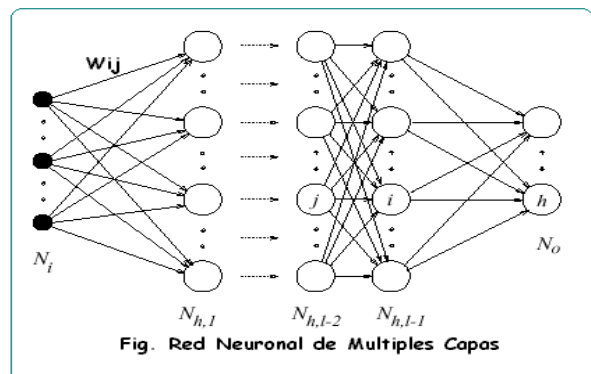
Otras aproximaciones con regresión han sido propuestas para encontrar coeficientes funcional:

- Jabbour usa una técnica de reconocimiento de patrones para mejorar 8 parejas horarias para un patrón de clima dado. Los correspondientes coeficientes de regresión lineal fueron usados.
- Una aplicación del algoritmo generalizado cuadrático lineal, fue propuesta por Irisarri. Sin embargo, a menudo arrastra inestabilidad numérica cuando se aplica a grandes bases de datos.
- Rahman, ha aplicado una aproximación de sistemas expertos. El sistema experto toma las ventajas del conocimiento del operador. Esto hace muchas divisiones de rango de temperatura y formas diferentes de relaciones funcionales de acuerdo a las horas de interés. Esto muestra bastante precisión del pronóstico. Según Tsoi, no es fácil extraer una base de conocimiento de un experto y puede mas bien ser difícil para el experto para articular su experiencia y conocimiento.

- Lu, utiliza el proceso modificado de ortogonalización de Gram-Schmidt, para encontrar un conjunto base ortogonal según el espacio de señal de salida formado por información de carga. Este proceso requiere una determinada cardinalidad del conjunto base ortogonal y el valor umbral de error usado en el procedimiento de adaptación. Si la cardinalidad del conjunto base es demasiado pequeño o el umbral no es suficientemente pequeño, la precisión de la aproximación sufre severamente. Por otro lado, si el umbral es demasiado pequeño, puede resultar una inestabilidad numérica. El proceso también tiene un problema de ambigüedad en la secuencia de los vectores de entrada. Diferentes exposiciones de vectores de entrada resulta en diferentes conjuntos de base ortogonal y diferentes pronósticos de salida.

3. Redes Neuronales Artificiales

3.1 Arquitectura



Una R.N.A. puede ser definida como un arreglo altamente conectado de elementos de procesamiento llamado neuronas. Un modelo ampliamente usado es el llamado Perceptron Multicapa. Una R.N.A. es mostrado en la figura. La R.N.A. tipo Perceptron Multicapa consiste de una capa de entrada, una o más capas ocultas y una capa de salida. Cada capa emplea varias neuronas y cada neurona en una camada es conectada a las neuronas en camadas adyacentes de pesos diferentes. El flujo de señales en la capa de entrada, pasa a través de la capa oculta, y termina en la capa de salida. Con excepción de la capa de entrada, cada neurona recibe señales de

las neuronas de las capas previas mediante valores o pesos entre las neuronas interconectadas. Las neuronas producen entonces señales de salida pasando una sumatoria a través de una función sigmoide.

Un total de Q conjuntos de datos de entrenamiento son requeridos. Entradas de $\{i_1, i_2, \dots, i_Q\}$ son impuestas en la capa inicial. La R.N.A. es entrenada para responder a los correspondientes vectores de entrada, $\{i_1, i_2, \dots, i_Q\}$, en la capa de salida. El entrenamiento continúa hasta que cierto criterio de parada es satisfecho. Típicamente, el entrenamiento es interrumpido cuando el error promedio entre la salida actual y la salida deseada de la red Neuronal sobre los Q conjuntos de datos de entrenamiento es menor que un predeterminado límite. El tiempo de entrenamiento requerido depende de varios aspectos incluyendo la complejidad del problema, el número de datos, la estructura de la red y los parámetros de entrenamiento usado.

3.2 Entrenamiento de la R.N.A

Se usará la regla delta generalizada para entrenar la R.N.A. tipo perceptron multicapa. Un vector de salida es producido presentando un patrón de entrada para la red. De acuerdo a las diferencias entre las salidas producidas, los pesos de la red $\{W_{ij}\}$ son ajustados para reducir el error de salida. El error de la capa de salida se propaga hacia atrás por la capa oculta, hasta que alcanza la capa de entrada. Debido a que el error se propaga hacia atrás, este procedimiento se conoce como algoritmo de retropropagación o "backpropagation".

La salida del neurón i, O_i , es conectada para la entrada del neurón j a través de los pesos de interconexión W_{ij} . A menos que k es uno de las neuronas de entrada, el estado del neurón k es:

$$O_k = f\left(\sum_i W_{ik} O_i\right)$$

Donde: $f(x) = 1/(1 + e^{-x})$, y la suma es sobre todas las neuronas en las capas adyacentes. Sea el estado destino de la neurona de salida t. Esto es, el error de la neurona de salida puede ser definida como:

$$E = \frac{1}{2}(t_k - O_k)^2$$

Donde k es el neurón de salida. El algoritmo de Gradiente descendente adopta los pesos de acuerdo al gradiente de error, esto es:

$$\Delta W_{ij} \propto \frac{-\partial E}{\partial W_{ij}} = \frac{-\partial E}{\partial O_j} \cdot \frac{\partial O_j}{\partial W_{ij}}$$

Específicamente, definimos la señal de error como:

$$\delta_j = \frac{\partial E}{\partial O_j}$$

Con algunas manipulaciones, obtenemos lo siguiente:

$$\Delta W_{ij} = \varepsilon \delta_j O_i$$

Donde ε es una ganancia de adaptación, δ_j es calculado basado en si la neurona j está ó no en la capa de salida. Si la neurona j es una de las neuronas de salida:

$$\delta_j = (t - O_j) O_j (1 - O_j)$$

Si la neurona j no está en la capa de salida,

$$\delta_j = O_j (1 - O_j) \sum_k \delta_k W_{jk}$$





A fin de mejorar las características de convergencia podemos introducir un término con ganancia de momento α :

$$\Delta W_{ij}(n+1) = \varepsilon \delta_j O_i + \alpha \Delta W_{ij}(n)$$

Donde n representa el índice de la iteración. Desde que la red Neuronal es entrenada. Esto solo requiere unas pocas multiplicaciones, adiciones y cálculos de la función sigmoide.

4. Aplicación de las Redes Neuronales Artificiales al Pronóstico de carga

La implementación de una RNA para pronostico de demanda envuelve los siguientes pasos:

-  Recolección de Datos de entrada
-  Selección de la arquitectura de la Red Neuronal
-  Entrenamiento de la Red
-  Prueba de la Red

- i) **Recolección de Datos de entrada**
Histórico de carga horaria para entrenar la red es necesario para realizar el pronostico. El histórico de datos puede incluir cargas horarias para el día actual y anterior; la carga pico y promedio de la ultima semana deberían también ser

usada. Otros datos de entrada a la RNA incluyen variables de entrada de temperatura y la estación. La temperatura es la variable más importante del clima. Debido al clima, a la geografía no uniforme en el área de servicio de la empresa eléctrica, y diferencias de temperatura en diferentes áreas se tendrán diferentes efectos en la carga del sistema. Es recomendable que la máxima temperatura pronosticada para el próximo día y la máxima temperatura actual registrada por diferentes áreas sea ingresada al modelo ANN.

ii) **Selección de la arquitectura de la Red Neuronal**

Las RNA son dispositivos multicapa. Muchas RNA constan de tres o más capas. La primera capa es la capa de entrada. Esta capa alimenta información del vector de entrada a través de los pesos de interconexión a la segunda capa, conocida como capa oculta. Podría haber más de una capa oculta. La capa final es la capa de salida. Mediante variación del número de capas ocultas y de elementos de procesamiento asociados por cada capa oculta para una particular aplicación. Los investigadores necesitan experimentar con diferentes arquitecturas para encontrar una adecuada solución. La experiencia demuestra que la red Neuronal de 3 capas parece ser la más exitosa configuración en el modelamiento de predicción de carga a corto plazo.

iii) **Entrenamiento de la Red**

Cada vector de entrada es enviado hacia adelante a través de la red, y un vector de salida es calculado. Durante el entrenamiento, la salida de la red es comparada con el actual dato histórico, y un término de error es creado. Este error es realimentado a través de la red, de la capa de salida, a hacia la capa oculta, y hacia atrás a la capa de entrada. Los pesos de interconexión entre cada capa son reajustados basados en el error calculado. Este proceso de presentar datos de entrada, pasarlos hacia adelante a través de la red y propagar el error hacia atrás es llamado la técnica de propagación hacia atrás (Backpropagation). El entrenamiento es continuado con los datos históricos mostrados a la red

muchas veces hasta que la tolerancia de error sea aceptable. Dependiendo de los datos que están siendo usados, el proceso de entrenamiento puede ser extenso y fácilmente alcanzar varias horas. Sin embargo, una vez que la red es entrenada, produce muy buenos resultados para una entrada de datos. Es importante saber que el sobreentrenamiento de la red puede resultar en memorización del patrón histórico de entrenamiento. Esto no es deseable, debido a que una red que ha memorizado el patrón de entrenamiento típicamente hace un pobre trabajo de generalización. Cuando nuevos patrones de datos son presentados para un sobreentrenamiento de la RNA, la salida producida tiende a parecerse al más cercano del conjunto de patrones memorizados.

iv) **Prueba de la red**

Cuando conseguimos el conjunto de datos de entrenamiento histórico, el investigador deberá guardar un pequeño conjunto de datos históricos. Datos que nunca han sido mostrados a la red durante el entrenamiento deberán ser usado para probar la red entrenada para medir su performance. Una vez que la red ha sido entrenada y probada para el más bajo error aceptable, la RNA puede ser usada para el modelamiento del pronóstico de carga.

Aspectos de los modelos de RNA :

Los nodos de las capas de entrada son hechos en base a los siguientes componentes que afectan la carga horaria:

- ❖ Bias
- ❖ Temperatura promedio de días pasados
- ❖ Temperatura de días pasados
- ❖ Temperatura pronosticada para futuras horas
- ❖ Horas del día
- ❖ Días de la semana
- ❖ Carga de horas pasadas

Durante el entrenamiento de una red, diferentes combinaciones son elegidas y probadas para asegurarse que el modelo fue continuamente refinado.

- Diferentes épocas de aprendizaje
- Diferentes conjuntos de Datos de Aprendizaje (secuencial o

- aleatorio) del entrenamiento a la red
- Diferentes conexiones (Full o no Full) entre nodos de procesamiento
- Diferentes números de nodos ocultos
- Diferentes funciones de transferencia (sigmoide, lineal, tangente hiperbólica)
- Diferentes esquemas de error (backpropagation)

5. Pruebas y Resultados

Para la implementación de la RNA se ha usado el ToolBox de Redes Neuronales de MatLab 5.3. Los datos de carga horaria han sido tomados del sistema inter-conectado Sur-Oeste, correspondiente a días laborables del año 1995. En vista que no se disponen de datos de temperatura estos fueron generados aleatoriamente. Se usan algunos datos para entrenar la RNA y otros para probar su performance. La Red Neuronal fue entrenada los siguiente cálculos:

- ❖ Carga Pico del día
- ❖ Carga total del día
- ❖ Carga Horaria

La estructura de la red Neuronal usada incluye el tamaño de la capa oculta, fueron elegidos de varias estructura. La estructura elegida es la que da mejor performance en términos de precisión. En muchos casos encontramos que añadiendo una o dos neuronas ocultas no hay un significativo efecto en la precisión de la red Neuronal.

5.1 Predicción de la Carga Pico

La topología de la R.N.A. para el pronóstico de la carga pico es como sigue:

- Neuronas de entrada : T1(k), T2(k) y T3(k)
- Neuronas ocultas : 5
- Neuronas de Salida : L(k)

Donde:

- K = Día de la Carga Pronosticada
- L(k) = Carga Pico del día K
- T1(k) = Temperatura promedio del día K
- T2(k) = Temperatura Pico del día K
- T3(k) = Temperatura menor del día K

5.2 Predicción de la Carga Total

La topología de la R.N.A.:

- Neurona de entrada : T1(k), T2(k) y T3(k)
- Neuronas ocultas : 5 Neuronas ocultas
- Neurona de salida : L(K)

Donde :

- k = Día de la predicción de la carga
- L(k) = Total de carga del Día K
- T1(k) = Temperatura Promedio del día k.
- T2(k) = Temperatura pico del día k.
- T3(K) = Temperatura más baja del día K

5.3 Predicción de la Carga Horaria

La topología de la R.N.A. para la carga horaria pronosticada con una hora de tiempo por delante es como sigue:

- Neuronas de Entrada : k, L(k-2), L(k-1), T(k-2), T(k-1) y T(x)
- Neuronas ocultas : 15 neuronas ocultas
- Neurona de salida : L(k)

k = Horas de predicción de Carga

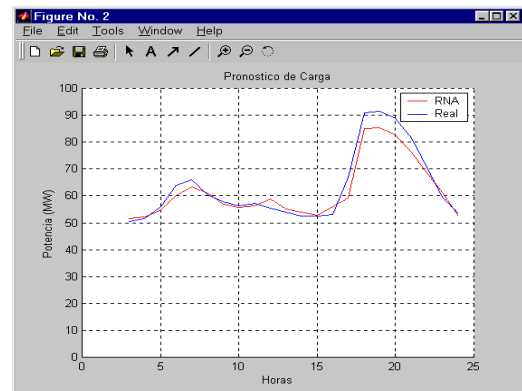
L(x) = Carga en la hora x,

T(x) = Temperatura en la hora x,

T(x) = temperatura proyectada para la hora x

En la etapa de entrenamiento, T(x) fue usado en lugar de T(x).

A continuación se muestra una simulación para un día laborable típico, para un juego de datos que no fue usado en el entrenamiento:



Se observan discrepancias debido a que no se disponen datos de temperatura real y se han tenido que asumir aleatoriamente. Trabajos realizados en el extranjero demuestran que el error es del orden del 1 o 2 %. Registrándose los mayores errores en las horas punta.

6. Conclusiones

Se ha presentado una metodología de pronóstico de carga usando una Red Neuronal Artificial. Los resultados muestran que la RNA es apropiada para interpolar entre datos de entrenamiento de patrones de carga y temperatura, para prever futuros patrones de carga.

A fin de pronosticar la carga futura con la RNA entrenada, necesitamos usar datos de carga y temperatura recientes y adicionalmente un pronóstico de la temperatura futura.

Los resultados se pueden considerar satisfactorios considerando que se han empleado datos temperatura aleatorios. De tener estos datos reales, obtendríamos la precisión del 1 o 2 % que se da en los trabajos de investigadores internacionales.

Comparado con otros métodos, la RNA permite una relación más flexible entre patrones de carga y temperatura.

Dado que la red Neuronal simplemente interpola los datos de entrenamiento, dará alto error con datos de prueba que no estén suficientemente cerca de algún dato de entrenamiento.

En general la Red Neuronal requiere grandes cantidades de datos de entrenamiento a fin de obtener resultados de alta precisión precisos.

El tiempo de entrenamiento requerido varía dependiendo del caso estudiado, desde algunos minutos a varias horas. Sin embargo, una red ya entrenada requiere solamente de algunos segundos para su prueba.

Solo se ha usado información de temperatura pero si se tuviera variables climáticas adicionales tales como nubosidad y velocidad del viento podría obtenerse mucho mejores resultados.

[3] A.Khotanzad, R.A. Rohani, D.Maratakulam, "ANNSTLF - Artificial Neural Networks Short-Term Load Forecaster Generation Three" IEEE transactions on Power Systems, Vol. 13, No. 4, November 1998, pp. 1413-1422.

[4] KD.Pham, "Load Forecasting Using Artificial Neural Networks" IEEE Conference Paper, 1995.

Referencias

[1] D.C. Park, M.A. El-Sharkawi, R.J. Marks, L.E. Atlas y M.J. Damborg, "Electric Load Forecasting Using An Artificial Neural Networks", IEEE transactions on Power Systems, Vol. 6, No. 2, May. 1991, pp. 442-449.

[2] C.N. Lu, H.T. Wu, S. Vemuri, "Neural Network based short term load forecasting", IEEE transactions on Power Systems, Vol. 8, No. 1, February. 1993, pp. 336-342.

Anexo

Programas Utilizados

```

Datos.m
% Datos de Carga - Sistema
Interconectado Sur Este
% Diagrama de Carga Promedio de dias
Laborables - Año : 1995 (MW)
L=[57.14 55.99 49.61 49.90 51.27 52.05
51.86 52.29 53.43 50.91 58.09 56.74;
55.53 54.39 48.02 48.50 50.40 50.80
50.73 51.29 52.21 50.33 56.87 55.92;
54.58 53.35 47.20 47.77 49.85 50.18
50.35 50.90 51.48 50.22 56.81 55.63;
55.13 54.03 47.31 48.22 50.19 50.75
51.74 52.16 52.39 52.35 58.98 57.38;
57.82 56.86 49.94 50.69 53.21 53.38
56.09 56.31 56.36 56.21 61.41 59.47;
60.10 59.55 52.77 54.96 58.02 58.44
63.93 63.15 60.08 61.62 67.75 63.83;
64.71 64.34 58.29 61.08 65.31 65.38
65.99 64.51 64.25 60.42 66.54 64.19;
63.37 63.54 57.26 56.70 58.81 59.81
60.28 58.29 58.39 56.17 63.31 62.35;
62.64 62.41 56.36 55.41 56.95 57.29
57.65 56.21 57.22 54.95 62.22 61.52;
62.30 61.60 55.46 55.18 56.16 56.64
56.43 56.36 56.57 55.20 62.10 61.30;
62.92 61.82 55.67 55.55 56.47 57.16
57.20 57.30 57.50 56.09 62.25 61.64;
62.45 61.26 55.06 54.91 55.33 55.53
55.36 55.58 56.81 55.06 61.75 60.78;
60.77 59.94 53.55 53.31 53.64 54.40
53.87 53.96 55.21 53.09 60.62 59.37;
58.95 56.74 51.57 51.19 52.56 53.26
52.62 52.42 54.11 52.28 59.22 57.90;
58.12 56.49 51.01 51.12 52.24 52.76
52.54 52.54 54.03 52.85 59.42 58.09;
58.21 57.27 51.39 50.69 51.63 52.46
53.06 52.92 55.02 54.72 61.16 59.21;
61.48 60.55 55.00 55.17 57.77 58.09
66.73 64.22 62.06 64.16 67.52 64.69;
61.43 60.44 65.57 74.98 82.53 82.26
90.97 85.48 83.98 82.87 87.73 83.87;
85.06 84.14 84.30 85.53 88.52 87.77
91.55 83.62 84.15 82.85 90.00 88.49;
85.60 83.57 82.58 85.15 85.86 85.39
88.84 85.56 85.12 82.80 89.28 87.62;
78.53 77.22 75.33 77.96 77.73 78.47
81.39 82.59 82.80 79.95 81.74 80.87;
66.65 66.47 65.15 67.56 66.92 68.34
70.73 72.13 71.67 69.06 70.48 70.72;
58.34 58.38 57.93 58.43 57.10 58.30
59.66 61.30 63.57 60.46 59.84 66.48;
58.27 57.91 51.18 51.14 52.53 53.26
53.30 53.77 56.13 53.97 61.16 59.95];
% Generacion aleatoria de datos de
temperatura
for i=1:12
    prome(i)=mean(L(:,i));
end
temp=[];
for i=1:24
    % temp=[temp; prome];
    temp=[temp; prome];
end
temp=temp/3;
temp=temp+(rand(24,12)-0.5)*2;

```

```

lfc.m
% Pronóstico de la carga por cada hora
(MW)
% Entradas : k, L(k-2), L(k-1),T(k-2),
T(k-1), T(k)
% Neuronas de la capa oculta : 15
% Salidas : L(k)
datos % Carga Datos
% Genera Datos de Entrenamiento
totent=[];
totsal=[];
for j=1:10
    for i=1:22
        ent=[i+2;L(i:i+1,j);temp(i:i+2,j)];
        sal=L(i+2,j);
        totent=[totent ent];
        totsalsal=[totsal sal];
    end
end
totent;
totsalsal;
net = newff([0 24; 0 100;0 100;0 30;0
30;0 30],[15 1],{'logsig' 'purelin'});
net.trainParam.epochs = 400;
figure(1)
net = train(net,totent,totsalsal);
% simulacion
figure(2)
j=7;
hold on
sacu=[]; lacu=[];
for i=1:22
    ent=[i+2;L(i:i+1,j);temp(i:i+2,j)];
    sal=sim(net,ent);
    l=L(i+2,j);
    sacu=[sacu sal];
    lacu=[lacu l];
end
plot(3:24,sacu,'r',3:24,lacu,'b')
axis([0 25 0 100])
grid
xlabel('Horas')
ylabel('Potencia (MW)')
title('Pronostico de Carga')
legend('RNA','Real')

```