

# Predicción de Ozono Mediante Redes Neuronales Heterogéneas

Jorge Orozco Luquero

Dept. Lenguajes y Sistemas Informáticos  
Universidad Politécnica de Cataluña Campus Nord,  
Módulo C6, Jordi Girona 1-3.  
08034, Barcelona, Spain.  
[jorozco@lsi.upc.es](mailto:jorozco@lsi.upc.es)

**Resumen** Obtener el máximo rendimiento de los datos experimentales es vital para obtener un modelo neuronal eficiente y fiable. Debido a la diferente naturaleza de los datos o a la existencia de valores perdidos, parte de la información contenida en las variables originales puede perderse en la adaptación y preprocesado. Las Redes Neuronales Heterogéneas permiten tratar directamente con esta información evitando el preprocesamiento. Esta estructura es comparada con otras redes feed-forward clásicas sobre un problema de predicción de ozono que presenta, entre otras cosas, valores perdidos. Los resultados muestran la capacidad de la HNN para efectuar la predicción sin un preprocesamiento previo. En algunos casos estos resultados mejoraron los obtenidos por las redes clásicas. En este sentido se demuestra que la introducción de información en la estructura de la red puede ser beneficiosa para mejorar los resultados en problemas con datos complicados de preprocesar.

## 1 Introducción

Una de las principales aplicaciones de las redes neuronales consiste en modelar sistemas complejos. Estos modelos son contruidos basándose en datos experimentales obtenidos del entorno del sistema. Por un lado, el rendimiento del modelo que se quiere construir se debe en gran parte a la relevancia y la fiabilidad de estos datos, y por otro a la propia capacidad y potencia de las técnicas de modelado empleadas. La adaptación, sin embargo, de los datos experimentales a un formato adecuado para la tarea puede hacer disminuir el rendimiento final del modelo. Con motivo de mantener al máximo la información inherente en los datos evitando traducciones y codificaciones, se plantea el uso de técnicas capaces de tratar directamente con datos provenientes de variadas fuentes.

Las Redes Neuronales Heterogéneas (HNN) [1] son un tipo de red feed-forward que se caracterizan por permitir atributos de muy diversos tipos. La tarea de codificación y adaptación previos al entrenamiento queda suprimida y la información contenida en los datos puede ser tratada directamente, incluso con valores perdidos. Esto ha de ser complementado con una mayor complejidad

en las neuronas de la red, que han de ser capaces de tratar adecuadamente con los distintos tipos de datos de entrada.

En el trabajo que se presenta a continuación se pretende predecir el comportamiento de la concentración de ozono en la Ciudad de México. En los datos utilizados se presentan valores perdidos, variables de tiempo discretas y mediciones experimentales, por lo que representan un grupo de atributos con una cierta heterogeneidad. Sobre este sistema se evalúan distintas técnicas de modelado: aquellas que necesitan un preprocesado frente a la HNN que no lo requiere. Dentro de las primeras se estudian en concreto el Perceptrón MultiCapa (MLP) y las Redes de Base Radial (RBF).

En estudios previos [2] se ha dividido este juego de datos en dos submodelos: uno para la temporada seca del año, cuando hay más actividad del ozono; y otro que trata de modelar el comportamiento en un mes concreto. En este estudio no se discute la eficacia de uno u otro submodelo, sino simplemente el rendimiento de la HNN frente a otras técnicas.

Para realizar la comparativa se han empleado técnicas de preprocesado de datos que incluyen el tratamiento de valores perdidos, frente a diversas configuraciones de redes heterogéneas capaces de tratar con variables discretas y valores con imprecisión. Las pruebas realizadas muestran un buen comportamiento de la HNN con respecto a las otras dos técnicas, obteniendo buenos resultados cuando se consideran variables como medidas con imprecisión.

El tratamiento de los datos con la HNN resulta más inteligente en el sentido en el que es posible establecer la función a calcular en las neuronas específicamente para cada tipo de atributo, introduciendo la información del sistema directamente en la red. La HNN otorga una gran flexibilidad para adaptarse a los datos, aunque sigue quedando abierta la elección de la función que calcula cada neurona. En sistemas muy específicos, un estudio detallado de las características de los datos y la capacidad de introducir esa información en la estructura de la red, pueden obtener con la HNN mejores resultados que técnicas no tan informadas acerca del dominio del sistema.

El documento se estructura tal y como se ha seguido la construcción del modelo. En primer lugar se introduce el concepto de las Redes Neuronales Heterogéneas y su diferencia con otras redes. A continuación se presenta la metodología seguida durante todo el proceso, con especial atención a cada uno de los pasos de preprocesamiento de los datos y el método de aprendizaje utilizado. Finalmente se presentan los resultados obtenidos y las conclusiones finales del estudio.

## 2 Redes Neuronales Heterogéneas

La Red Neuronal Heterogénea define la neurona como una unidad de cálculo de similitud. Dentro de esta definición se encuentran incluidas las neuronas de un MLP y una RBF, las cuales se consideran casos particulares de funciones de similitud homogéneas. Este cálculo se realiza entre dos vectores (el de entradas  $\mathbf{x}$  y el de pesos  $\mathbf{w}$ ) específicos para cada neurona.

Este nuevo modelo permite obtener más información acerca de la estructura interna de la red. Debido a que las neuronas calculan funciones de similaridad, es posible introducir conocimiento previo del dominio en la estructura de la red. La tarea de las neuronas está de este modo claramente delimitada y el proceso de interpretación es mucho más sencillo porque existe una semántica precisa para el cálculo neuronal.

Formalmente, las funciones de similaridad se definen como una función  $f$  en un espacio heterogéneo  $\mathcal{H}$ , donde  $f : \mathcal{H}^n \times \mathcal{H}^n \rightarrow \mathcal{H}$ . Esta función  $f$  puede verse como una generalización de las funciones calculadas en neuronas clásicas como el MLP o las RBF. De hecho, si se considera que  $f$  tiene la siguiente estructura:

$$f(\mathbf{x}, \mathbf{w}) = g(h(\mathbf{x}, \mathbf{w})) \quad (1)$$

donde  $n$  es la dimensión de los pesos y entradas,  $\mathbf{x}$  es el vector de entradas y  $\mathbf{w}$  el vector de pesos. Considerando que  $\mathcal{H} = \mathbb{R}$ , se establecen los valores de las funciones  $g$  y  $h$  de tal modo que se consigue una neurona MLP si  $g$  es una función sigmoideal y  $h$  un producto escalar y una neurona RBF si  $g$  es una función gaussiana y  $h$  una distancia euclídea.

Existen por lo tanto muchas posibles funciones que pueden considerarse similitudes siempre que cumplan unas adecuadas condiciones [1]. La información del dominio del tipo de dato está incluida en esta función de similaridad que es preciso definir adecuadamente.

Aparte de la heterogeneidad de las neuronas, del vector de entradas y el vector de pesos, la estructura de una HNN es idéntica a la de otras redes feed-forward comunes. Las neuronas que calculan funciones de similaridad son las que residen en la capa oculta mientras que las de salida actúan como integradoras del resultado. Las entradas al ser heterogéneas, a diferencia del resto de redes, pueden aceptar datos de muy distintos tipos, destacando:

**Nominales** Conjunto finito de variables discretas y sin orden.

(e.g. nombres de ciudades).

**Ordinales** Como las nominales, pero ordenadas.

(e.g. "Lunes", . . . , "Viernes").

**Continuas** Números reales. Entradas comunes de las redes neuronales.

**Números difusos** Representantes de imprecisión en torno a un valor central llamado *moda*.

(e.g. medidas imprecisas  $5.4 \pm 0.05$ ).

**Intervalo difuso** Variables lingüísticas, normalmente tienen orden.

(e.g. "caliente", "frío", "templado").

**Valores perdidos** No es realmente un tipo de dato sino que es la falta de datos.

La HNN acepta y trata directamente con los valores perdidos.

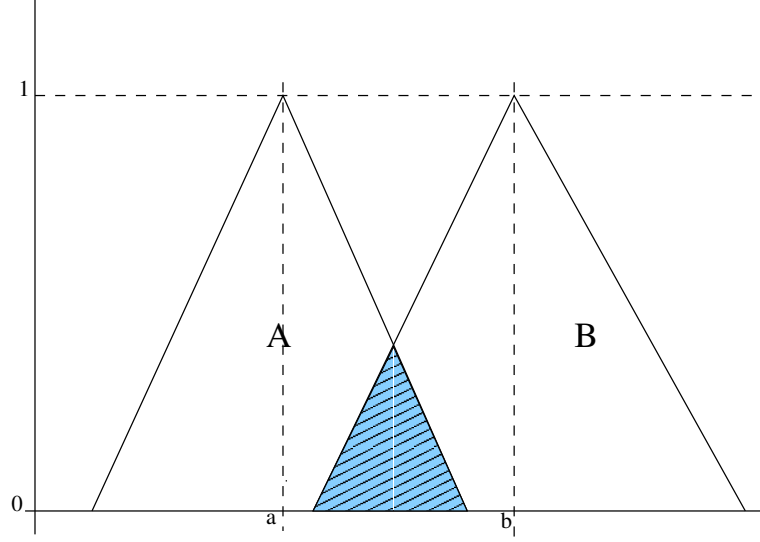
En este caso concreto que se estudia se presentan variables ordinales, continuas y difusas. Las ordinales pueden ser de dos tipos: *lineales* o *circulares*. La primera se define como aquella en la que la distancia entre los valores extremos de la escala es la máxima. Su similaridad se define como

$$s(x_k, y_k) = 1 - \frac{|x_k - y_k|}{\text{rango}(k)} \quad (2)$$

donde  $rango(k)$  es el número de valores de la variable  $k$ .

La variable ordinal circular se define como aquella en la que la distancia entre los valores extremos de la escala es mínima y positiva (e.g. los días de la semana). Una similitud posible para este tipo de variable es:

$$s(x_k, y_k) = \max(1 - \frac{|x_k - y_k|}{rango(k)}, \frac{|x_k - y_k|}{rango(k)}) \quad (3)$$



**Figura 1.** Intersección de números difusos

Para el caso de las variables difusas la similitud se realiza en función de la intersección y la unión entre los números difusos:

$$s(x_k, y_k) = \frac{I(x_k, y_k)}{U(x_k, y_k)} \quad (4)$$

La similitud entre variables continuas se expresa de la siguiente manera:

$$s(x_k, y_k) = \hat{s}(|x_k - y_k|) \quad (5)$$

Donde  $\hat{s}$  es una función continua y estrictamente decreciente en  $[0, 1]$  definida como:

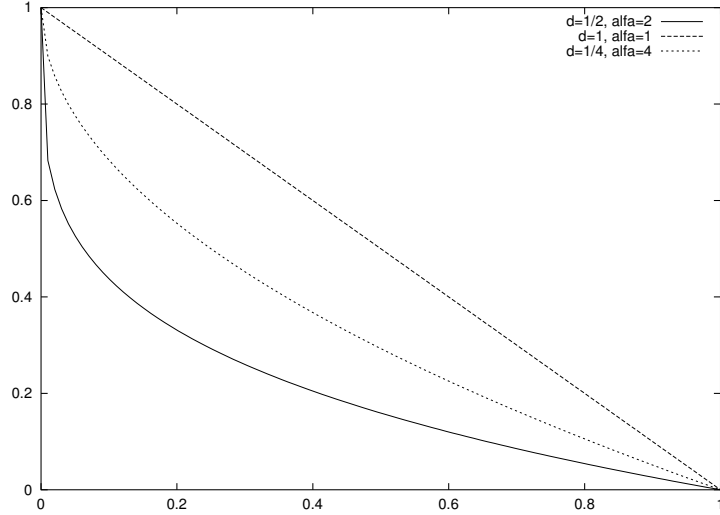
$$\hat{s}(z) = (1 - z^d)^\alpha \quad (6)$$

donde  $z$  ha de ser una distancia.

Los parámetros  $d$  y  $\alpha$  se utilizan para regular la sensibilidad de la neurona a la distancia calculada. En este estudio se han probado las siguientes configuraciones (ver fig. 2.):

- $d = 1; \alpha = 1$
- $d = \frac{1}{2}; \alpha = 2$
- $d = \frac{1}{4}; \alpha = 4$
- $d = \frac{1}{4}; \alpha = 1$

Otras muchas variantes se e las cuales se estudian en [1]. Es preciso buscar los parámetros anteriores adecuados para cada tipo de problema.



**Figura 2.** Diferentes configuraciones de  $\hat{s}$

En cuanto al algoritmo de entrenamiento en las HNN, debido a la existencia de dominios no reales, la derivabilidad de la función de error no está clara. Por lo tanto, técnicas de descenso por gradiente como Back-Propagation no son aplicables en este caso. Se utiliza por ello un algoritmo genético denominado Breeder Genetic Algorithm (BGA) [3].

### 3 Metodología

Lo que se dispone a realizar a continuación es obtener un modelo preciso para poder predecir la concentración de ozono en un momento determinado en base a otras variables obtenidas del entorno. Para ello se parte de tres redes básicas: MLP, RBF y HNN. De las cuáles se prueban diversas configuraciones de neuronas y parámetros de entrenamiento. En especial, de la HNN se evalúa el rendimiento al usar diversas combinaciones de tipos de datos y funciones  $\hat{s}$ .

La metodología a seguir en el proceso es la siguiente. En primer lugar es necesario utilizar la técnica de time-delay para adaptar los datos obtenidos de la

serie temporal del ozono a una red feed-forward como la HNN. A continuación es conveniente realizar una selección de variables para simplificar el modelo y eliminar variables redundantes. Como paso previo al entrenamiento, es necesario adaptar los datos al formato de las redes y se establecen los parámetros estructurales (e.g. número de neuronas ocultas) y de entrenamiento que se van a probar. Finalmente se comenta el procedimiento utilizado para entrenar y para seleccionar las mejores configuraciones de parámetros.

### 3.1 Selección de variables

La adquisición de los datos se realiza previamente al caso que se estudia, por lo que el primer punto del diseño se centra en la selección de variables relevantes. Existen diversas técnicas para realizar esta tarea como los algoritmos incrementales o decrementales. La teoría de Rough Sets[4] parte de la idea de identificar la cantidad de información contenida en las variables, buscando obtener el mínimo conjunto perdiendo la mínima información. Otras técnicas de modelado contienen mecanismos propios de obtención de dependencias como la herramienta de Razonamiento Inductivo Difuso (FIR) [5].

En el caso concreto que se trata en este trabajo, los datos han de ser adaptados para que la red feed-forward sea capaz de encontrar dependencias temporales. La técnica que se ha utilizado ha sido la de time-delay, que consiste en añadir nuevos atributos desfasados una serie de unidades de tiempo con respecto a otros. Los inconvenientes de usar la técnica de time-delay son, por un lado, la estimación de la longitud temporal máxima existente entre variables. Por otro, la explosión dimensional que estas dependencias producen sobre la entrada. En lo relativo al primer punto, estudios previos en el caso del ozono [6][2] muestran dependencias temporales de menos de tres pasos previos. Debido al aumento de la dimensión de la entrada, se hace más necesario si cabe el proceso de selección de variables.

Para este trabajo, y en vista de resultados previos en el mismo campo, se parte de una expansión en la entrada de cuatro valores previos por variable (i.e. además del valor de la variable en  $t$ , ahora se dispone del valor en  $t-1$ ,  $t-2$ ,  $t-3$  y  $t-4$ ). Originalmente se parte de 7 atributos: *Hora*, *Día de la semana*, *Velocidad del viento*, *Dirección del viento*, *Temperatura*, *Humedad relativa* y *Medición de Ozono*. Se obtienen por lo tanto  $7 \cdot 4 = 28$  variables. Aplicando al resultado la técnica de Rough Sets [4] se obtienen los siguientes conjuntos reducidos de variables para los dos submodelos a tratar:

- Modelo de Temporada (Enero - Mayo 2000)

$$(Ho(t-2), Di(t-2), Ve(t), Oz(t-2), Oz(t-1)) \rightarrow Oz(t)$$

- Modelo de Mes (Enero 1996-2000)

$$(Ho(t), Di(t), Ve(t), Oz(t-2), Oz(t-1)) \rightarrow Oz(t)$$

### 3.2 Preprocesamiento

Las redes neuronales usualmente tratan los datos como valores continuas. Sin embargo, muchos sistemas tienen variables discretas, con orden establecido o contienen más información que la que simplemente almacena una variable real (e.q. valores con imprecisión). Generalmente estos atributos se traducen a un formato continuo alterando su estructura y significado. Los valores perdidos, por ejemplo, suelen codificarse añadiendo un atributo más que establece si en un patrón concreto aparece o no un valor perdido. Esta alteración de la estructura de los datos oscurece el proceso interno que realiza la red impidiendo obtener información en este sentido.

En el caso de las HNN estas traducciones no son necesarias, ya que puede manejar directamente este tipo de información. En este estudio se han comparado varios de estos métodos por separado evaluando el rendimiento individualmente. En el estudio que se presenta los datos contienen estas características:

- En los datos del ozono existen un 30% de valores perdidos en la variable de *Velocidad del Viento*.
- La *hora* y el *día de la semana* se consideran variables discretas y circularmente ordinales.
- Las mediciones de Velocidad del Viento y Ozono se considera que contienen imprecisión de medida y son tratadas como difusas.

Los valores perdidos en el MLP y RBF se codifican con una variable más que con un valor de 1 indica que la Velocidad en ese patrón es desconocida y con un 0 lo contrario. La hora y el día se consideran en estas redes valores continuos al igual que el resto de variables.

### 3.3 Entrenamiento de las redes

Debido al manejo de información heterogénea y a la dificultad de obtener la derivabilidad de una hipotética función de error, la HNN se entrena mediante el algoritmo BGA. Sus parámetros principales son el número de generaciones, el tamaño  $\mu$  inicial de la población y el porcentaje de población seleccionada para recombinarse  $\tau$ . Este algoritmo también parametriza la recombinación y la mutación. Es importante destacar que estos operadores genéticos también han de ser adaptados al dominio y ser específicos para cada tipo de dato. Se utiliza en este proceso la técnica de recombinación Extended Intermediate Recombination (EIR) y Mutación Continua (CM).

En el caso que se estudia se utiliza una configuración de parámetros que ha resultado ser útil en otras optimizaciones de Redes Neuronales[7]. Los valores elegidos son:

Tamaño inicial ( $\mu$ )	60
% recombinación ( $\tau$ )	25
EIR con parámetros ( $\delta$ )	25
CM con parámetros ( $\rho$ )	0.5

Para certificar el correcto entrenamiento se emplea un mecanismo de k-fold Cross Validation (CV) en el que se ha establecido  $k$  al valor de 5. Los resultados obtenidos son producto de la media de 10 ejecuciones por cada partición realizada por CV.

A continuación es necesario obtener la selección de modelo neuronal a utilizar. El número de neuronas ocultas y los parámetros de la función  $\hat{s}$  que transforma la similitud  $(d, \alpha)$  son obtenidos seleccionando el mejor de entre varias ejecuciones de CV con diversas configuraciones. A continuación se presenta el procedimiento de pruebas realizado y los resultados obtenidos.

### 3.4 Procedimiento

Tratando de aislar cada una de las características especiales de la HNN se realizan diversos experimentos de contraste. En la primera fase se evalúa el rendimiento de las redes clásicas MLP y RBF con distintas configuraciones neuronales en ambos submodelos (Modelo de Temporada y Modelo de Mes). A continuación se ejecutan diversas configuraciones de neuronas y valores de  $d$  y  $\alpha$  sobre la HNN solamente con codificación directa de Valores Perdidos. En esta primera fase se obtienen los mejores valores de número de neuronas y función  $\hat{s}$  con respecto a los resultados obtenidos.

En la segunda fase se ejecuta la HNN considerando la *Hora* y el *Día de la Semana* como variables ordinales circulares, mientras que los valores perdidos se codifican de la manera clásica. También se realiza esta operación considerando la *Velocidad del Viento* y las mediciones de *Ozono* como variables difusas con un 0.5% de imprecisión. Finalmente se realizan dos últimas pruebas combinando la técnica de Ordinales con codificación directa de Valores Perdidos por un lado y las tres técnicas (Ordinales, Valores Perdidos y Números Difusos) por otro.

El criterio de error utilizado en el experimento es el Error Cuadrático Medio Normalizado (NMSE) y se calcula de la siguiente manera:

$$NMSE = \frac{\sum (output - target)^2}{N * var} \quad (7)$$

Donde  $var$  es la varianza de la salida de la red.

Ante estas configuraciones cabe esperar un rendimiento mejor de la HNN con Valores Perdidos debido a la menor dimensionalidad. Hay que considerar, sin embargo, que las otras redes pueden utilizar esa variable extra para intentar reducir el error. Por otra parte, al restringir dos variables a números discretos se está delimitando la capacidad de ajuste de los pesos y esto puede también repercutir en el resultado.

## 4 Resultados

En la siguiente tabla se muestran los valores obtenidos en test para los distintos modelos y configuraciones neuronales utilizadas. Para el MLP se selecciona una red neuronal de 4 neuronas ocultas, mientras que la RBF funciona mejor con 8.

Por su parte la HNN que mejor resultado obtuvo fue la que utilizó 8 neuronas ocultas y la siguiente funcion  $\hat{s}$ :

$$\hat{s} : f(d) = 1 - \sqrt[4]{d} \quad (8)$$

En la tabla 1 pueden verse las distintas versiones de HNN ejecutadas (Codificación de valores perdidos, ordinales, difusos y la combinación de estas técnicas). Se presentan dos tablas, una para cada uno de los modelos que se ha utilizado.

**Tabla 1.** NMSE obtenido con redes clásicas y varias configuraciones de Redes Heterogéneas. Se presentan los resultados de los dos submodelos

Modelo de Temporada					
Tipo Red		Num. hidden	NMSE	$d$	$\alpha$
<i>MLP</i>		4	0.2471	-	-
<i>RBF</i>		8	0.3184	-	-
<i>HNN</i>	V.P.	8	0.2625	1/4	4
	ORD.	8	0.4031	1/4	4
	FUZ.	8	0.2542	1/4	4
	V.P. & ORD.	8	0.6259	1/4	1
	V.P. & FUZ.	8	0.2920	1/4	1
	V.P. & ORD. & FUZ.	8	0.2795	1/4	1

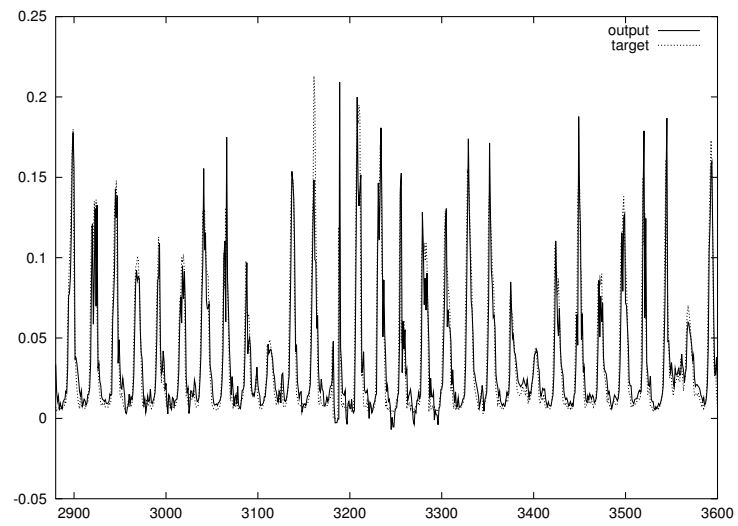
  

Modelo de Mes					
Tipo Red		Num. hidden	NMSE	$d$	$\alpha$
<i>MLP</i>		4	0.7632	-	-
<i>RBF</i>		8	0.3685	-	-
<i>HNN</i>	V.P.	8	0.3030	1/4	4
	ORD.	8	0.2481	1/4	4
	FUZ.	8	0.2344	1/4	4
	V.P. & ORD.	8	0.2537	1/4	4
	V.P. & FUZ.	8	0.2412	1/4	1
	V.P. & ORD. & FUZ.	8	0.2662	1/4	1

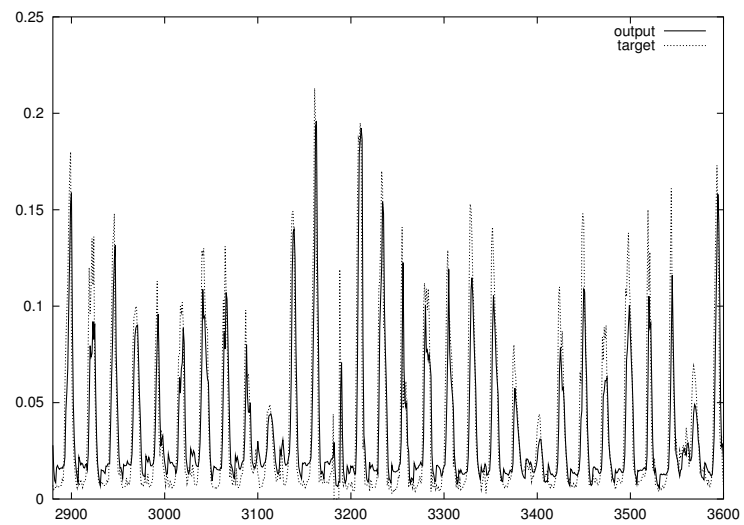
Los resultados se ilustran en las figuras 3, 4 y 5, el ajuste realizado por la mejor red MLP, RBF y HNN en una porción del conjunto de datos.

## 5 Conclusiones

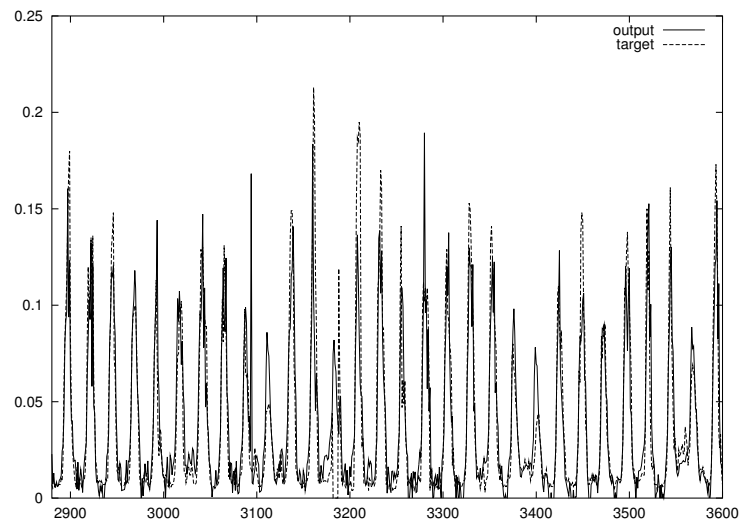
A la hora de realizar un modelo computacional basado en datos experimentales aparecen graves inconvenientes desde la misma adquisición de los datos. Estos en ocasiones son incompletos o imprecisos y no se ajustan al formato de las herramientas de análisis comunes. Por eso mismo exigen un preprocesamiento previo que en muchos de los casos puede limitar la cantidad de información de los datos, poniendo en riesgo el buen rendimiento del modelo. Al contrario de los métodos tradicionales que realizan la codificación de los datos para ajustarlos a



**Figura 3.** Predicción de parte del modelo de temporada con MLP



**Figura 4.** Predicción del mismo modelo con RBF



**Figura 5.** Una HNN con V. Perd. y Variables Difusas

su formato, las Redes Neuronales Heterogéneas permiten tratar directamente con datos continuos, ordinales, nominales, lingüísticos e incluso valores perdidos. En el caso de las redes que trabajan con continuos esto no es posible y, en concreto, en el caso de los valores perdidos, necesitan aumentar el número de variables y alterar la codificación.

En este trabajo se ha presentado un estudio para modelar el comportamiento del Ozono en la Ciudad de México. Estos datos simplemente sirven de prueba para comprobar de un modo práctico la incidencia que tiene el uso de heterogeneidad en los datos con respecto a la homogeneidad. Concretamente, los datos presentados contienen variables ordinales, imprecisas y valores perdidos; y así han sido tratados por la HNN en distintas variantes.

Los resultados obtenidos contrastados con el MLP y la RBF muestran que en ambos casos se ha conseguido mejorar el rendimiento, sobre todo en el caso de la RBF. La mejora, aunque ligera, es significativa ya que confirma que el hecho de introducir conocimiento dentro de las estructuras de la red o los datos puede conducir a mejores resultados. Hay que notar, sin embargo, que no todas las versiones de la HNN consiguieron mejorar resultados. Se ha visto, por ejemplo, que las variables ordinales restan flexibilidad a la red obteniendo menores resultados. Ello puede ser debido a la restricción que se aplica a las variables ordinales discontinuas, limitación que no posee el MLP, por ejemplo. El tratamiento de valores perdidos obtiene en general también buenos resultados aunque en algún caso la mayor dimensionalidad en la entrada puede beneficiar al MLP debido a que existen un mayor número de parámetros libres en el sistema y le otorga mayor flexibilidad. Sin embargo, la combinación de técnicas, especialmente la de números difusos y valores perdidos, obtiene los mejores resultados obtenidos

hasta ahora. De hecho, estudios recientes en este campo obtienen resultados similares [6][2].

Por todo ello se puede concluir que el hecho de manipular mínimamente la información puede beneficiar a la red, siempre que se introduzca en las neuronas la información adecuada para tratar correctamente los datos. Esta capacidad puede ser un arma de doble filo en el sentido de que da mayor flexibilidad a la hora de elegir la función que calculan las neuronas, pero también exige una búsqueda de la función más adecuada para el problema concreto.

## Agradecimientos

Este trabajo ha sido subvencionado por el Consejo Interministerial de Ciencia y Tecnología (CICYT), bajo el proyecto TAP1999-0747 y por el Ministerio de Educación, Cultura y Deporte (MECD) con la beca FPU AP2001-3735.

## Referencias

1. Belanche, L.: Heterogeneous Neural Networks: Theory and Applications. PhD thesis, Universitat Politècnica de Catalunya, Barcelona, Spain. (2000)
2. P. Gómez, A.Ñebot, F.M.: Fuzzy reasoning for the prediction of maximum ozone concentration. In: 13th European Simulation Symposium. (2001) 535–542
3. Mühlenbein, H., Schlierkamp-Voosen, D.: Predictive models for the breeder genetic algorithm. *Evolutionary Computation* **1** (1993) 25–49
4. Ohrn, A.: ROSETTA Technical Reference Manual. Knowledge Systems Group, Department of Computer and Information Science (NTNU), Trondheim (Norway). (2000)
5. Klir, G.: Architecture of Systems Problem Solving. Plenum Press (New York) (1985)
6. G. Acuña, H.J., Perez, R.: Neural network model for maximum ozone concentration prediction. In: Proceedings in the International Conferences on Artificial Neural Networks (ICANN). (1986) 263–268
7. Belanche, L.: Similarity-based heterogenous neuron models. In Horn, W., ed.: ECAI'2000: 14th European Conference on Artificial Intelligence, IOS Press (Amsterdam) (2000)