

## REDES NEURONALES PARA LA DISCRIMINACIÓN DE NUBES EN EL PROCESO DE CÁLCULO DE TEMPERATURAS SUPERFICIALES

J.A. Triñanes, J. Vázquez, M. Martínez y J. Arias  
Laboratorio de Sistemas. Dpto. Electrónica e Computación. Facultad de Física. Campus Sur. 15706  
Santiago de Compostela. E-mail: eljoki@usc.es

### RESUMEN

La identificación de las nubes es fundamental en la extracción de mapas de temperatura superficial oceánica. Los métodos usuales están basados en técnicas de umbral. En este artículo presentamos un método basado en redes neuronales que a nuestro entender tiene una mayor flexibilidad que los anteriores. Se describe también el algoritmo de entrenamiento, que produce una rápida convergencia hacia el mínimo de error.

### INTRODUCCIÓN

Para la extracción de nubes de imágenes SST se han empleado diversas técnicas. El método tradicional está basado en la aplicación sucesiva de umbrales a una serie de tests sobre los pixels de la imagen. Sólo aquellos pixels que pasan todos los tests entran en el proceso de extracción SST. Estos tests suelen diferir en algún punto según la imagen sea diurna o nocturna y los umbrales se determinan de manera empírica. Además, la mayor parte de ellos son tests de coherencia espacial que tienen en cuenta el entorno del pixel sujeto a estudio. Los resultados obtenidos mediante estas técnicas de umbrales son relativamente buenos pero el método es demasiado rígido y es difícil mantenerlo en una posición de equilibrio entre las tendencias de eliminar por exceso o quedarse corto y mantener pixels nubosos que pueden causar problemas en un posterior tratamiento de la imagen. Es por esta razón que hemos abordado este problema desde otra perspectiva. Las redes neuronales son un método muy atractivo puesto que tienen la capacidad de "aprender" patrones cuya complejidad hace difícil definirlos mediante aproximaciones más formales, como resulta en este caso. Es por ello que también se revelan muy útiles para clasificar los diversos tipos de nubes.

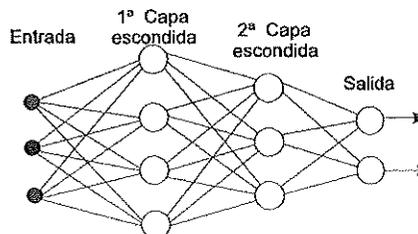
### REDES NEURONALES

Las redes neuronales han sido ampliamente usados en los últimos años en la clasificación de imágenes procedentes de sensores multispectrales[1]. Los modelos basados en redes neuronales tienen la ventaja respecto a los modelos estadísticos en que no se necesita un conocimiento previo de la distribución

estadística de los datos de entrada. En cada caso, cada modelo de red neuronal está caracterizado por su topología interna y su regla de aprendizaje. El modelo más representativo del aprendizaje supervisado es el perceptrón multicapa (MLP) con el algoritmo de aprendizaje denominado back-propagation o regla delta generalizada. El modelo MLP es, sin duda alguna, el modelo de red neuronal más popular. Ha sido empleado en una gran número de problemas de procesamiento de información como reconocimiento de patrones, control no lineal, predicción de series de tiempo caóticas,... Todos los nodos (o neuronas) de la red, excepto los de la capa de entrada, llevan a cabo dos funciones: recoger las salidas de los nodos de la capa anterior y establecer una salida no lineal. Así, considerando un vector de entrada n-dimensional. Cada neurona produce una suma ponderada de cada uno de los componentes del vector y añade un término de desplazamiento o umbral que puede ser tanto positivo como negativo y que puede ser considerado como un nodo en estado permanente de activación. El resultado se emplea como argumento de una función no lineal que establece la salida de la neurona. Estos dos procesos se pueden reflejar en la siguiente fórmula[2],

$$y_i^p = f \left( \underbrace{\sum_j \omega_{ij}^{p-1} y_j^{p-1}}_{x^p} + \theta_i^p \right)$$

donde  $y_i^p$  es la salida de la neurona  $i$  situada en la capa  $p$ ,  $f$  es la función no lineal,  $\omega_{ij}^{p-1}$  es el peso entre este nodo y otro  $j$  situado en la capa anterior  $p-1$  cuyo estado es  $y_j^{p-1}$  y  $\theta_i^p$  es el valor de umbral.



Figural. Esquema de una perceptrón multicapa.

## ALGORITMO DE APRENDIZAJE

El algoritmo de aprendizaje que nosotros empleamos se denomina backpropagation o regla delta generalizada y es un método de entrenamiento supervisado donde la corrección del error se lleva a cabo mediante la minimización de la suma del cuadrado de los errores o error cuadrático medio (ECM) de los vectores de salida,

$$ECM = \frac{1}{2} \sum_p \sum_i (y_{i,p}^H - o_{i,p})^2$$

donde p indexa los patrones de entrada e i las salidas,  $y_{i,p}^H$  es la respuesta obtenida en la salida i de la capa H de salida para el patrón p y  $o_{i,p}$  es el nivel de salida esperada. Para cada patrón de entrada, se ajustan las conexiones de la red de tal manera que la respuesta de los nodos de salida se iguale al valor del patrón de salida. Los conjuntos de entrenamiento se presentan a la red repetidamente hasta que ésta "aprenda" los patrones y ajuste los pesos de manera conveniente. Recordemos que los pesos representan una medida del grado de correlación entre la salida o niveles de actividad de las neuronas que conectan. Los pesos se suelen inicializar a valores aleatorios muy pequeños[3] y en cada iteración se determinan de acuerdo a,

$$\omega_{ji}(t) = \omega_{ji}(t-1) - \varepsilon \left. \frac{\partial ECM}{\partial \omega_{ji}} \right|_{\omega(t)}$$

donde  $\varepsilon$  es una constante positiva denominada factor de aprendizaje. El problema radica en encontrar una expresión que nos permita calcular la derivada parcial de ECM respecto a cada peso de la red. Para ello y a partir de las ecuaciones anteriores,

$$\frac{\partial ECM}{\partial \omega_{ji}} = \frac{\partial ECM}{\partial y_j} \frac{dy_j}{dx_j} \frac{\partial x_j}{\partial \omega_{ji}} = \frac{\partial ECM}{\partial y_j} y_j^p (1 - y_j^p) y_j^{p-1}$$

Pero en el nodo de salida obtenemos una condición de contorno donde la sensibilidad de las neuronas en la última capa se deriva de la ecuación del ECM,

$$\frac{\partial ECM}{\partial y_j} = y_j^H - o_j$$

Para las otras capas,

$$\frac{\partial ECM}{\partial y_j} = \sum_k \frac{\partial ECM}{\partial y_k} \frac{dy_k}{dx_k} \frac{\partial x_k}{\partial y_j} = \sum_k \frac{\partial ECM}{\partial y_k} \frac{dy_k}{dx_k} w_{kj}^h$$

donde j y k indexan las capas h y h+1 respectivamente.

La sensibilidad de cada nodo se computa desde la capa de salida hacia atrás. Resumiendo, la idea central es computar los niveles de todas las neuronas de la red para cada patrón de entrada-salida. Seguidamente se realiza un proceso en sentido contrario en el que, comenzando en la capa de salida, se computa la derivada del error

$$\frac{\partial ECM}{\partial y_j} \text{ y se propaga hacia atrás para}$$

actualizar los pesos hasta que alcanzamos la capa de entrada[4].

Para intentar adaptar el factor de aprendizaje a la curvatura total de la superficie de error se ha añadido un término de momento  $\alpha$  a la actualización de cada peso para acelerar el aprendizaje de la red, con  $0 < \alpha < 1$ . Así, la actualización de los pesos quedaría,

$$\omega_{ji}(t) = \omega_{ji}(t-1) - \varepsilon \left. \frac{\partial ECM}{\partial \omega_{ji}} \right|_{\omega(t)} + \alpha \Delta \omega_{ji}(t-1)$$

Este término provoca que la dirección de búsqueda sea una media ponderada exponencialmente de direcciones pasadas, y ayuda a que los pesos cambien en porciones planas de la superficie de error. El momento elimina oscilaciones y permite que el factor de aprendizaje aumente sin provocar inestabilidades, acelerando el entrenamiento de la red.

## CONSIDERACIONES

La mayor debilidad del modelo MLP radica en la extrema lentitud del proceso de aprendizaje. El ECM describe una superficie en el espacio de pesos. Los algoritmos de aprendizaje podrían considerarse como métodos que buscan un mínimo en esta superficie y la naturaleza de la misma gobierna la complejidad del proceso. Para MLP, las superficies de error tienen características bien conocidas que hacen la búsqueda de un mínimo difícil. Por ejemplo, tienen regiones llanas en las que el aprendizaje es lento y otras zonas que son llanas en una dirección y con fuertes desniveles en otras. Para acelerar el aprendizaje emplearemos un método adaptativo de ajuste del factor de aprendizaje y

control del momento[5]. Consiste básicamente en incrementar  $\epsilon$  si en el último paso el error total ha disminuido. En caso contrario, inhabilitamos el momento y reducimos el factor de aprendizaje. La razón estriba en que  $\alpha$  está dirigiendo la búsqueda en la dirección incorrecta y debemos reducir  $\epsilon$  para encontrar la dirección apropiada. Cuando el error disminuya de nuevo, incluiremos el momento otra vez e incrementaremos el factor de aprendizaje con cada paso favorable.

Es peligroso inicializar  $\epsilon$  a valores muy altos ya que los pesos se pueden saturar en pocas iteraciones produciéndose oscilaciones en el sistema. Para evitar esta situación, hemos escogido como valor de inicio un valor inversamente proporcional al número de patrones de entrenamiento y al número de nodos de la red.

### ENTRENAMIENTO

La elección de los conjuntos de entrenamiento es uno de los aspectos en los que debemos tener más cuidado. El entrenamiento de la red debe realizarse sobre pixels que correspondan a la categoría establecida como salida. En caso contrario el aprendizaje será incorrecto y los resultados de la clasificación final estarán degradados por efecto de la mala selección de los patrones de entrada. Asimismo debemos seleccionar un conjunto de patrones, denominado de test, que serán introducidos en la red entrenada para validar los resultados de la clasificación. Las salidas de la red sobre el conjunto de patrones de test son una medida del rendimiento del sistema y nos sirven para comparar con otros métodos estadísticos o de umbral que también se suelen emplear para discernir la cobertura nubosa.

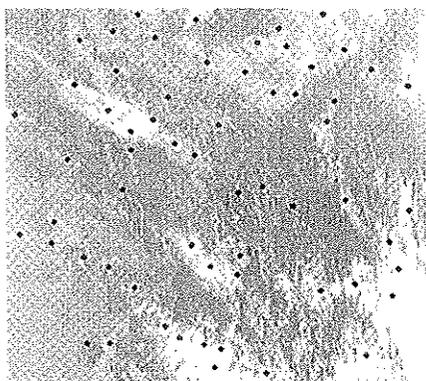


Figura 2. Imagen AVHRR, canal 4. Hemos superpuesto algunos conjuntos de entrenamiento.

A la hora de eliminar las nubes con los criterios basados en umbral, algunos de los tests dependen de los valores de los pixels vecinos. En ocasiones, las nubes de subresolución provocan cambios en la temperatura de brillo del pixel que sólo pueden ser evaluados comparándolos con los puntos del entorno. Resulta claro entonces que para mejorar los resultados de nuestra red necesitamos introducir medidas de textura. Con el propósito de mejorar la fiabilidad del método hemos entrenado la red neuronal en función de tres características: zona geográfica, satélite y hora de pase. Rutinariamente, en nuestro laboratorio procesamos imágenes de diversas zonas del mundo. Para tener en cuenta las peculiaridades de cada una de ellas, hemos entrenado una red diferente según trabajemos, por ejemplo, en el Océano Índico o en el Atlántico Norte. La discriminación según satélite resulta obvia. En función de la hora de paso también modificamos la naturaleza de las entradas a la red. No tiene sentido emplear las bandas 1 y 2 durante la noche, del mismo modo que el ruido presente en la banda 3 aconseja apartarla del proceso de detección de nubes para las imágenes diurnas. Así, por ejemplo, para imágenes nocturnas, las entradas de la red son los niveles digitales de las bandas 3, 4 y 5, así como las diferencias  $ND_4 - ND_3$ ,  $ND_3 - ND_5$  y  $ND_4 - ND_5$ . Otra entrada son la máxima diferencia en el canal 4 dentro de una ventana de  $3 \times 3$  respecto al pixel central.

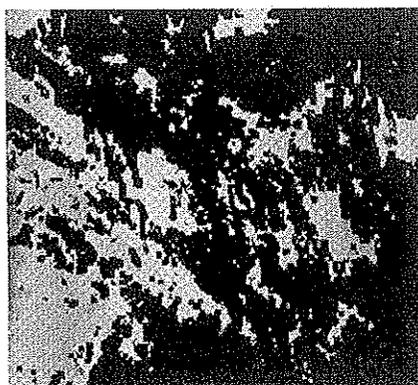


Figura 3. Imagen SST una vez extraídas las nubes.

Para imágenes diurnas hemos optado por la introducción de los niveles digitales de todas las bandas del sensor menos la 3, que en esta situación tiene mucho ruido. También introducimos las diferencias entre las bandas 1 y 2, y 4 y 5. Como en las imágenes nocturnas, introducimos una medida de la relación con los

pixels vecinos a través de la diferencia máxima entre el pixel central y los valores de sus vecinos en un entorno 3x3.

La simplicidad de los datos de entrada del discriminador es esencial para asegurar rapidez de procesamiento cuando se aplique sobre la imagen. Otro factor a tener en cuenta es que para imágenes diurnas debemos eliminar los efectos que diferentes ángulos cenitales del sol tienen sobre los niveles digitales de la imagen recibida. Para ello dividimos el nivel digital por el coseno del ángulo anterior. En imágenes nocturnas esto no es necesario y el sistema automáticamente descarga el proceso anterior de la secuencia de procesamiento.

## CONCLUSIONES

Los resultados obtenidos son muy esperanzadores. Desde nuestro punto de vista, los resultados han mejorado respecto a los métodos tradicionales de eliminación de pixels nubosos. Se están llevando esfuerzos en cuantificar la diferencia entre este y otros métodos que presentaremos en el futuro. Entre las principales ventajas de nuestro método resaltamos la gran capacidad de adaptación y la facilidad de entrenamiento de la red. Además, la sencillez de la misma facilita el procesamiento en tiempo real de las imágenes recibidas.

## BIBLIOGRAFÍA

- [1] J.A. Triñanes, J. Torres, A. Tobar y C. Hernández, "Clasificación de Imágenes Multiespectrales Mediante Redes Neuronales", *Revista de Teledetección*, 3:46-50. Nov (1994).
- [2] J.A. Benediktsson, P.H. Swain y O.K. Ersoy, "Neural Network Approaches Versus Statistical Methods in Classification of Multisource Remote Sensing Data", *IEEE Trans. Geosci. Remote Sens.*, 28:540-551 (1990).
- [3] D. Hush y B. Horne, "Progress in Supervised Neural Networks", *IEE Signal Processing Magazine*, Enero 1993.
- [4] S.K. Pal y S. Mitra, "Multilayer Perceptron, Fuzzy Sets, and Classification", *IEEE Trans. Neural Net.*, 3:683-697 (1992).
- [5] P.D. Heermann y N. Khazenie, "Classification of Multispectral Remote Sensing Data Using a Back-Propagation Neural Network", *IEEE Trans. Geosci. Remote Sens.*, 30:81-88 (1992).