

IMPLEMENTACIÓN DE UNA RED NEURONAL AUTOORGANIZATIVA PARA EL ANÁLISIS HIPERESPECTRAL SOBRE PROCESADORES DSP

J.M. Sánchez-Dehesa, I. García, P.L. Aguilar, M.C. Cantero

chema@unex.es

Departamento de Informática. Universidad de Extremadura

ABSTRAC:

El presente trabajo trata del análisis de imágenes hiperespectrales usando modelos de computación neuronal no supervisada (SOM) y su implementación sobre procesadores DSP. La red SOM ha sido ampliamente probada con estos datos [ref. pepe] y, dada su elevada complejidad computacional, se propone una implementación alternativa cuyo objetivo fundamental es reducir el tiempo de cómputo. La complejidad computacional es del orden de $O(N_x M_x B_x S_x I)$, donde $N_x M$ son las dimensiones de la imagen a procesar, B es el número de bandas, S son las neuronas de salida e I es el número de iteraciones. Este valor para una imagen de 86×83 con 224 bandas es del orden de 800M. Actualmente los procesadores DSP están a la cabeza de la tecnología para este tipo de operaciones, además su arquitectura está especialmente optimizada para el procesamiento paralelo y la respuesta en tiempo real [Garc2003]. Las últimas generaciones de procesadores DSP incluyen herramientas para la programación paralela, y pueden alcanzar una velocidad de proceso de 900 MIPS. La implementación del algoritmo de red neuronal se ha realizado sobre la plataforma DSK C6000 de T. I., usando las herramientas software que dicha plataforma incluye para la programación paralela. El análisis de los resultados de tiempo de ejecución comparativamente con una implementación sobre un computador clásico, nos lleva a concluir que los procesadores DSP pueden ser una solución alternativa al cómputo clásico sin apenas coste adicional y obteniendo una respuesta mucho más cercana al tiempo real.

KEYWORDS: dsp, hiperespectral, kohonen, TI

1. INTRODUCCIÓN

Las Redes Neuronales (RN) son sistemas bio-inspirados formados por elementos de proceso simples (neuronas), distribuidas en capas. Y han sido usadas con éxito para el tratamiento hiperespectral, debido principalmente a su robustez frente al ruido, a la caracterización de las entradas y su rapidez si se implementa de forma paralela dado que existe un elevado número de elementos de proceso que implica un alto grado de paralelismo, también existe facilidad

para obtener a partir de una red neuronal una implementación paralela [Pere98]; además, el procesamiento de información en una RN involucra operaciones matriciales.

De los diferentes modelos de computación neuronal destacan las redes autoorganizativas, caracterizadas por no requerir de ningún agente externo que actúe de supervisor en el proceso de aprendizaje (Kohonen, 1997). La red autoorganizativa de Kohonen ha demostrado ser una herramienta muy útil para la clasificación de patrones y más concretamente para la de imágenes hiperespectrales [Agu2000]

Los procesadores DSP han irrumpido en el mercado con fuerza en los últimos años haciéndose un hueco entre los computadores tradicionales y los específicos (ASIC) que tuvieron su momento en los años 80. Esta expansión en su uso se debe a las características que hacen que confluyan en él las ventajas del ordenador clásico (programabilidad) y del específico. Así, los procesadores DSP están especialmente capacitados para el procesamiento digital de señal, por ejemplo puede realizar un instrucción MAC (multiplicación y acumulación) en un solo ciclo de reloj, o disponen de memorias separadas para datos e instrucciones, paralelismo interno (pipeline), etc. Las últimas generaciones disponen de una arquitectura de CPU dividida en varias unidades que pueden trabajar en paralelo, pudiendo obtener un rendimiento de hasta 8 instrucciones ejecutándose en paralelo [TMS2000]. Por otra parte incorporan herramientas software cuya función es la de soportar las operaciones actuando como una capa de S.O en tiempo real que gestiona y distribuye los recursos entre las distintas tareas, según el tipo y prioridad de cada tarea.

Haciendo uso de todos los recursos que los actuales procesadores DSP ponen al alcance, se ha realizado una implementación modular, escalable y flexible del algoritmo de red SOM sobre el DSK C6000. Para ello se parte de una implementación en lenguaje C con las clásicas funciones que se van ejecutando de forma secuencial hasta concluir el tratamiento. A raíz del estudio de la estructura del algoritmo original y de las restricciones impuestas por la dependencia de los datos, se definen las tareas (threads TSK) para la

nueva aplicación, así como los módulos encargados de la transferencia de datos entre las tareas [TMS 2002].

2. RED NEURONAL SOM

A partir del conocimiento de la formación biológica de mapas que se da en el cerebro, es decir, el hecho de que los grupos de neuronas especializadas en una determinada función se sitúan próximas, T. Kohonen idea un sistema de red neuronal inspirado en este comportamiento [Koho82]. Se trata de una red con capacidad de formar *mapas de características* de forma similar a lo que ocurre en el cerebro. El objetivo de Kohonen es *demonstrar que un estímulo externo (información de entrada) por sí solo, suponiendo una estructura propia y una descripción funcional del comportamiento de la red, es suficiente para forzar la formación de los mapas* [Koh89].

La red de Kohonen o SOM es una red *off-line*, esto es, existe una fase de aprendizaje y otra de funcionamiento. La arquitectura se presenta en la siguiente figura 1:

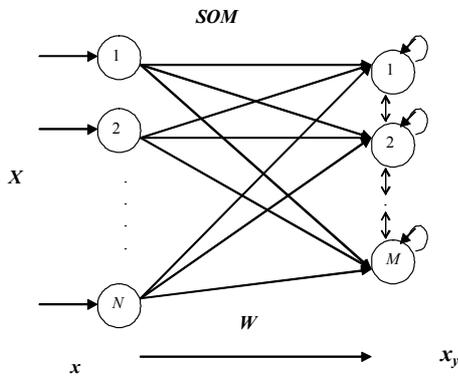


Figura 1.- Topología de las Redes SOM

Esta red neuronal consta de los siguientes elementos:

- Una capa de N neuronas de entrada, lineales, es decir dejan pasar la información de entrada (X)
- Una capa de M neuronas de salida, competitiva.
- Un conjunto de conexiones hacia delante, donde cada una de ellas lleva asociado un vector de pesos (w_i).
- Un conjunto de conexiones laterales y autorrecurrentes en la capa de salida que producen efectos inhibitorios y excitatorios de las mismas.

En la fase de aprendizaje la red aprende las características del conjunto de entrenamiento (X); en esta fase se pueden distinguir tres etapas que se repiten hasta que el mapa esté completamente formado: *muestreo*, *coincidencia* y *actualización*. En la fase de funcionamiento se propaga la entrada (x) de la capa de entrada a la de salida, se computa su salida (Wx) y, aquella neurona con máxima salida, determina la clase a la cual pertenece la entrada.

3. ESTRUCTURA DEL ALGORITMO DE RED NEURONAL PARA CLASIFICACIÓN DE IMÁGENES

El algoritmo de red neuronal se puede dividir en una serie de pasos que realizan las etapas de la fase de aprendizaje. Estos pasos se van sucediendo en secuencia, de forma que cada procedimiento requiere los datos del procedimiento anterior. La figura 2 muestra un esquema del algoritmo donde quedan reflejadas las tres etapas del aprendizaje (muestreo, coincidencia y actualización), así como la dependencia de datos entre ésta

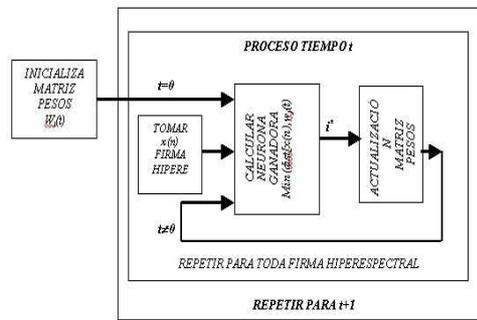


Figura 2.- Estructura del algoritmo de red SOM

Según se desprende de la figura anterior las tareas *calcularneuronaganadora* y *actualizacionmatrizpesos* son interdependientes en el sentido de que cada uno de ellos requiere los datos de salida del otro. No ocurre lo mismo con la tarea de muestreo, que es independiente del resto de tareas. Esta tarea respecto al bloque formado por las otras dos presenta una estructura de *pipeline*, y por tanto fácilmente paralelizable, esto es, esta tarea puede ser ejecutada a

la vez que el segundo bloque correspondiente a las dos tareas; mientras la red calcula neurona ganadora y actualiza pesos para una determinada firma hiperspectral, se puede estar tomando la siguiente firma que va a ser tratada. Esta no es una cuestión menor, pues actualmente la información de la imagen se lee de un dispositivo de memoria secundaria, proceso bastante costoso desde el punto de vista del tiempo.

El algoritmo de red es un proceso repetitivo en dos estadios, el proceso se repite para cada firma hiperspectral $x(n)$ y para cada iteración t . En todas las iteraciones la matriz de pesos utilizada es la resultante de la última actualización.

4. IMPLEMENTACIÓN DEL ALGORITMO EN LA PLATAFORMA DSK C6000 DE T.I.

El algoritmo ha sido implementado previamente en C para comprobar su funcionamiento y obtener los primeros resultados de tiempo de ejecución. Se ha realizado únicamente la fase de *aprendizaje*, puesto que ésta es la más compleja, y la fase de *funcionamiento* es idéntica a una de las etapas del aprendizaje.

En la implementación de la red SOM sobre la plataforma C6000 existen distintos niveles de refinamiento que nos llevan a una solución cada vez más óptima. De todas las posibilidades aquella que utiliza los recursos DSP/BIOS, es la que mejores resultados proporciona pues, con esta herramienta y en función del tipo de módulos seleccionados, la tarea actual no tiene que concluir exactamente para que otra

5. ANÁLISIS DE RESULTADOS

La complejidad computacional del algoritmo es del orden de $O(N \times M \times B \times S \times I)$, donde $N \times M$ es la dimensión de la imagen a procesar, B sería el número de bandas de frecuencia, S el número de neuronas de salida e I es el número de iteraciones hasta la convergencia. Este valor para una imagen pequeña de 83×86 con 224 bandas es del orden de 800M. De este valor se deduce que los requerimientos de tiempo y memoria para la ejecución del algoritmo son muy elevados.

Nuestra implementación ha mejorado sustancialmente el tiempo de cómputo del algoritmo, aunque sin llegar a una respuesta en tiempo real por las razones expuestas en el apartado anterior. La figura 3 en muestra los resultados de tiempo para los procedimientos de cálculo de la neurona ganadora y actualización de matriz de pesos. El resultado mejora

sea llamada por el sistema, es lo más parecido a procesamiento paralelo, cualquier slot de tiempo de la CPU será aprovechado en la ejecución de las tareas que conforman el programa.

Evidentemente no es el objetivo de este trabajo obtener respuesta en tiempo real para nuestro algoritmo, por varias razones:

1. Las imágenes están almacenadas en un fichero, aunque en el futuro puede plantearse que estas se estén emitiendo desde un punto determinado y puedan ser recogidas en tiempo real.
2. El coste computacional es excesivamente elevado para una imagen relativamente pequeña, y para computar en un solo procesador DSP

Pero sí puede ser un objetivo el optimizar ese tiempo (varias horas) para hacerlo más adecuado a las necesidades actuales.

En la implementación del algoritmo de red sobre la plataforma DSP se han hecho uso de los recursos para la programación paralela, esto, el kernel DSP/BIOS, que permite diseñar aplicaciones con características de modularidad, escalabilidad y flexibilidad. Se definen módulos que pueden ser ejecutados de forma concurrente, con la única restricción de la dependencia de los datos y de las prioridades asignadas en el proceso de configuración. La figura 3 muestra el fichero de configuración utilizado para la implementación del algoritmo de red. El tipo de módulos (threads) es el de tareas TSK, ya que dadas sus características de ejecución y prioridad es el que mejor se adapta a nuestra aplicación.

un orden de magnitud el tiempo de ejecución obtenido en un computador de propósito general. Pero este tiempo puede ser mejorado si se adecuan las características de la placa al tipo de tratamiento, esto es, mayor capacidad de memoria y adaptador para la captura de imagen (en lugar de audio que es el que actualmente dispone la plataforma).

6. CONCLUSIONES

Este trabajo presenta una implementación alternativa para el algoritmo de red SOM, para ello se ha utilizado la plataforma DSK C6000 de T.I. El procesador DSP C6000 pertenece a una nueva generación de procesadores que permiten diseñar aplicaciones con las características de modularidad, escalabilidad y flexibilidad, y en las cuales los módulos (threads) que integran la aplicación pueden ser ejecutados de forma concurrente con la única restricción de la dependencia de los datos, a partir de

propiedades como prioridades, etc. Con esta implementación se ha mejorado ostensiblemente el

tiempo de cómputo requerido por la aplicación, de forma que sola la lectura de la imagen de un fichero

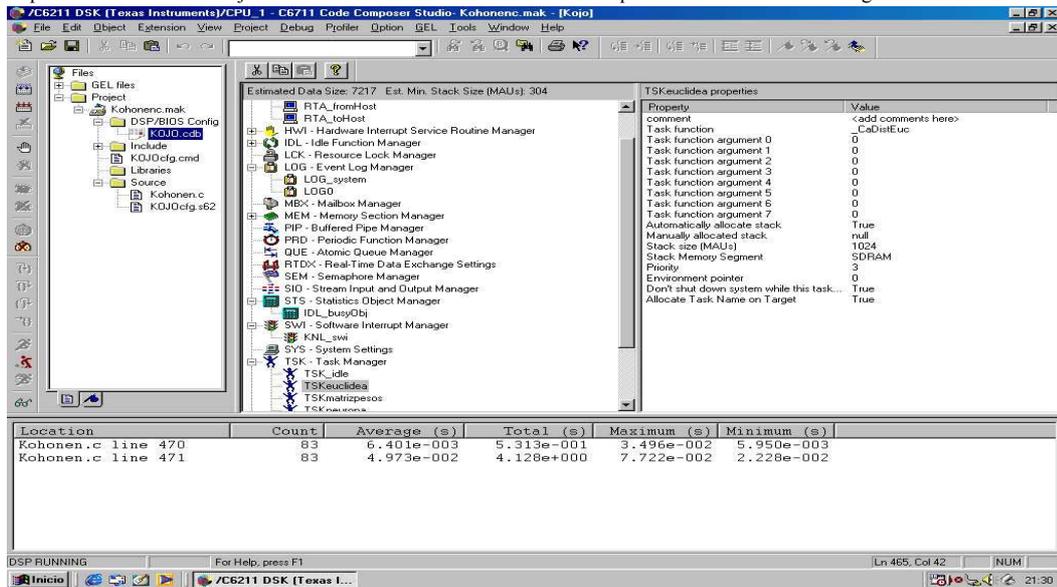


Figura 3- Resultados de tiempo de calculo

retrasa el proceso en su conjunto. Además la plataforma utilizada esta ideada para proceso de señal de audio y, por lo tanto no dispone de una entrada adecuada de señal, ni de la memoria necesaria para poder contener la imagen a procesar. De todo esto se deduce que la mejora de tiempo obtenida se debe exclusivamente a la arquitectura de CPU del DSP, y que esta reducción podría ser mucho mayor si se adecuan las conexiones y la memoria al tipo de datos que se están tratando.

Este trabajo se ha realizado dentro del proyecto 2PRI01A085 cuyo investigador principal es Pablo Martínez.

7. REFERENCIAS

[Garc2003] I. García, A. Cordero, J.M. Sánchez-Dehesa and P. Gomez, "A real time implementation of voice recognition system based on DSK Platform TMS320C6000", ISPC Conference Proceedings, Dallas, Abril 2003.

[Pere98] Pérez R.M., Bachiller P., Martínez P. And Aguilar, P.L. "Nn-Quantizer for solving the mixture problem and its implementation by systolic Arrays", Microelectronics Journal, Vol.30,pp.77 82, 1998

[Agui2000] Aguilar P.L., Plaza A., Martínez P. and Pérez R., "Endmember extraction by a self-organizing neural networks on hyperspectral images", Proc. International Conf. On Automation, Robotics and ComputerVision, Nanyang Technological Institute, Singapore, 2000.

[koho82] T. Kohonen, "Self-Organized formation of topologic correct Feature Maps", Biological Cybernetics, Vol. 43, pp 59-69, 1982.

[TMS 2000] TMS320C6000 DSP/BIOS User's Guide, Texas Instruments, May 2000

[TMS 2002] TMS320C6000 Programmer's Guide, Texas Instruments, 2002 www.ti.com