CHAPTER 8

SYSTOLIC S.O.M. NEURAL NETWORK FOR HYPERSPECTRAL IMAGE CLASSIFICATION

P. Martínez, P.L. Aguilar, R.M. Pérez and A. Plaza

Departamento de Informática, Universidad de Extremadura
Campus Universitario s/n 10071 Cáceres, Spain
E-mail: pablomar,paguilar,rosapere,aplaza @unex.es

Hyperspectral image sensor developments on the study of the Earth's surface give way to images with higher spectral and spatial resolutions. In fact, the higher the resolution, the greater the size of these images. The use of these sensors by space-borne satellite systems will provide an enormous and continuous flow of data with constraints placed on onboard storage, and data transmission bandwidth. New algorithms and computer capabilities will be necessary for the classification, compression and pre-processing of these images. In this chapter, we propose an SOM algorithm for hyperspectral classification implemented by one systolic array, to provide real-time hyperspectral compression facilities.

8.1. Introduction

The use of remote sensors to study the Earth canopy has created a great deal of expectations for applications in very different fields.

Some of these applications are not possible for the low spatial and spectral resolution of first-generation remote sensors.

Currently, hyperspectral sensors lack these restrictions, providing a large number of narrow bands that ensure sensor capabilities for narrow absorption band recognition.

Molecules and particles in land, water and atmosphere environments interact with the sun energy in the 400-2500nm spectral region through absorption, reflection and scattering processes; these occur in narrow spectral regions. Hyperspectral sensors are developed to measure spectra as images in these portions of the spectrum [1, 2] .

The amount of hyperspectral image (HI) spectral bands results in large size data sets; for example, a DAIS image with a size of 614x512 spatial pixels occupies about 140 Mbytes.

The high spatial and spectral resolution of HIs can be used for high resolution classification and/or determination of the Earth's surface constituent signature variation. This is achieved via the hyperspectral information provided by the sensor for a composite pixel spectrum. These analytical capabilities increase the HI applications.

Recent scientific research and applications are expanding HI investigations of ecology and vegetation, geology and soils, inland and coastal waters, the atmosphere, snow and ice, hydrology, biomass burning, environmental hazards [2].

Today, most hyperspectral images are taken from airborne sensors. Each image datum requires one careful schedule of the flight, regional weather, flight permission...etc. Future space-borne satellite hyperspectral sensors will provide access to spectral images from all regions of the Earth with multi-temporal coverage.

The new sensors will present significant challenges to the measurement of high quality spectra from space.

One of the bottlenecks of these new sensors is the enormous and continuous flow of data that is produced [3]. Data compression becomes increasingly important in these applications for two reasons:

- Onboard storage.
- Data transmission bandwidths.

New special computer architectures must be designed to aid massive compression tasks. This paper discusses the systolic implementation of one unsupervised classification algorithm performed by an SOM neural network.

The design is based on the contiguous flow of multidimensional data, using the parallel capabilities of the systolic arrays to provide real time computing facilities.

8.2. SOM Neural Network for Hyperspectral Analysis

Most algorithms in conventional multi-spectral images cannot be used with hyperspectral formats due to the following reasons.

1. Great size of the hyperspectral images (30 times larger than a Landsat TM image of the same spatial size).

2. High dimensionality of the images.
3. New Sub-pixel analysis possibilities.
4. High discrimination capabilities to resolve classes
5. Difficulties to use training data.

New methods should be applied for hyperspectral image classification; they should consider the mentioned features, and should be capable of exploiting both high dimensional feedback and spatial information.

The use of neural networks techniques for hyperspectral image processing have increased in the past few years.

The neural network approach has the following advantages [4]:

- Simplicity.
- A lot of parameters for adjusting performance.
- Intuitive method
- Intrinsic parallelism (different degrees of freedom for implementations)
- Easy VLSI implementation.
- Viable for high dimensional data.

Unsupervised classification algorithms do not require ground-truth data. They are namely interesting in remote sensing due to the great cost of these data in Earth canopy studies.

Unsupervised clustering leads to problems with applications in many areas. Given a set of N data points in a feature space of D dimensions $(x1,x2,x3.....xD) \in RD$, we wish to characterize the data as belonging to a K cluster, where K must be obtained from the statistics of the image data without ground truth information. This clustering is based on distance metrics. Various unsupervised algorithms can be used to classify hyperspectral data: ISODATA, K-means. The accuracy of these unsupervised algorithms is usually very low [5].

One of the most useful neural network (NN) algorithms is the Self-Organizing NN, or Self Organized Map (S.O.M), proposed by T. Kohonen, with applications in various signal processing tasks. The SOM neural network is an unsupervised classification method, widely used for image treatment. One of the most interesting SOM characteristics is the creation of topologic maps [6].

Some reasons for using S.O.M. in hyperspectral analysis have been described by [7]:

- Avoiding the need to degrade data

- Providing speed (when implemented in hardware as massively parallel algorithms).
- Surpassing conventional classification algorithm performance.
- Good performance for large real-life tasks.

In the following section, we design one S.O.M. to classify HIs.

8.3. Topology of the Proposed Neural Network

Our proposed network architecture is depicted in Fig. 8-1 [8]. In our case, N corresponds to the number of channels of the hyperspectral image, and M is the number of classes or prototypes to be extracted by the network. M depends on image complexity and must be carefully selected according to certain metrics. The weight matrix W has one weight for the connection of each input neuron (channel) with each prototype neuron.

In the classification phase, the input signals x are projected on the feature space by the feed-forward connections W; each neuron produces a selective response to the input signals. In the learning phase, lateral and feedback output layer connections produce excitatory or inhibitory effects depending on the distance from the neuron to the winning neuron [6]. These weights are used to determine the Wi classification prototype for each output neuron.

8.4. SOM Training Algorithm

There are five basic steps involved in the training algorithm. These steps are repeated until the topological map is completely formed:

a) Initialization. Choose random values for the initial weight vectors $w_i(0)$, i = 1,2,...,M. It is desirable to keep the magnitude of the weights small.

b) Sampling. Choose an input pattern $x(n)$ belonging to the pixel of the hyperspectral image. The selection is done randomly.

c) Similarity Matching. Find the best-matching (winning) neuron i^* at time t, using the minimum-distance criterion, as shown in the following equation:

$$i^*[x(n)] = \min_j dist\{x(n), w_j(t)\} \qquad j = 1,2,\cdots,M \qquad (8\text{-}1)$$

where $dist\ (i^*, i)$ is the Euclidean distance.

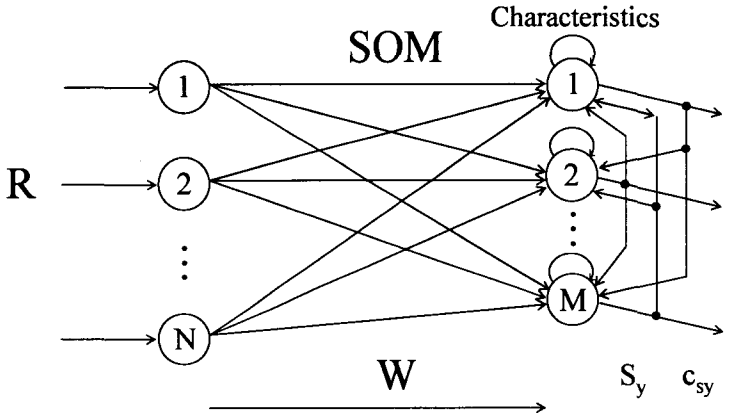

Fig. 8-1. SOM neural network topology including weight matrix W and lateral and feedback connections.

d) Learning. Adjust the synaptic weight vectors of all neurons, using the update formula (8-2), where η(t) is a learning-rate parameter, and γ is a Gaussian neighborhood function centered around the winning neuron. The size of the neighborhood is determined by a parameter σ(t) (see Eq. (8-3)).

$$w_i(t+1) = w_i(t) + \eta(t)\gamma(t,i,i^*[x(n)])(x(n) - w_i(t)) \tag{8-2}$$

Among the different choices made for the involved parameters, we follow the approach in [9], thus coming up with:

$$\eta(t) = \frac{1}{t} \qquad \gamma(t,i,i^*[x(n)]) = e^{\frac{dist(i^*,i)^2}{\sigma(t)}} \qquad \sigma(t) = \left(\frac{\sigma_0}{t}\right)^2 \tag{8-3}$$

where σ_o is the initial width.

e) Continue from step b) until no noticeable changes in the weight space are observed, or until the maximum convergence time is achieved.

8.5. Systolic Algorithm

To alleviate the SOM learning algorithm described above, one solution is the connection of a large number of identical single processing elements (PEs). Each PE has a private storage, is arranged in one structure, and is connected only to the neighboring PE's. This structure is referred to as Systolic Array [10, 11].

The systolic array provides the following advantages:

- Exploits naturally the segmentation of regular networks with local connections.
- Produces high performance, decreasing the communication cost.
- Offers a good balance between computation and communication for array operations.

In order to design the systolic linear array for the implementation of the SOM algorithm described above, the first step is the use of one regular algorithm with local dependences, equivalent to the algorithm described in Section 4:

```
For j=1 to M
   s(j,0)= 0
   For  i=1 to N
        x(0,i)=x_i
        x(j,i)=x(j-1,i)
        s(j,i)= s(j,i-1)+(x(j,i)-w(j,i)) *(x(j,i)-w(j,i))
   End For
End For
```

From these sequential algorithms, we can obtain the dependence graph with connected MxN nodes, as shown in Fig. 8-2 [12].

In order to establish the correspondence between the dependence graph and the corresponding systolic algorithm, we develop a bijection T, going from the original algorithm index set to the systolic index set:

$$T = \begin{bmatrix} 11 \\ 10 \end{bmatrix} \tag{8-4}$$

According to this bijection, we should have one graph projection in the east-west direction. For these reasons, the difference between the input pattern

and the weight vector s(j) of the neuron j will remain static for the corresponding EPj. The input pattern will be displaced from one EP to its right neighbor. The total number of cycles in the retrieving phase will be M+N.

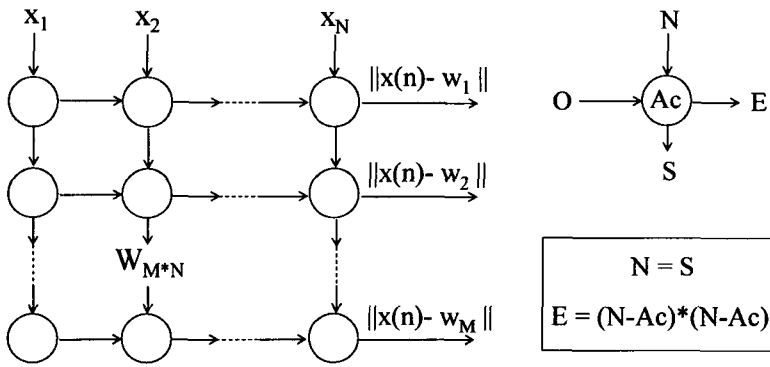

Fig. 8-2. Dependence graph to compute similarity between the input pattern and the weight vectors.

The data movements within the systolic algorithm for distance computation, are the following:

- The input data *x(n)* are introduced in the first EP and are propagated to the other EPs sequentially.
- The weight data are stored in the EPs row by row.
- When the *x(n)* value arrives at the j EP, the square of the difference between the stored weight on the EP and the *x(n)* is computed, and the partial sum s_j is stored on the same EP.
- The control unit changes the EP mode to read its weight in the following step, and is compared with the other s_j to obtain the winning neuron.

Now, we describe the components of each EP:

- Memory: Each EP stores one row of the weight matrix.
- Communication: The data moves along one direction between neighbor EPs (E=W).
- Processing: Each EP supports all arithmetic operations including subtraction and accumulation.

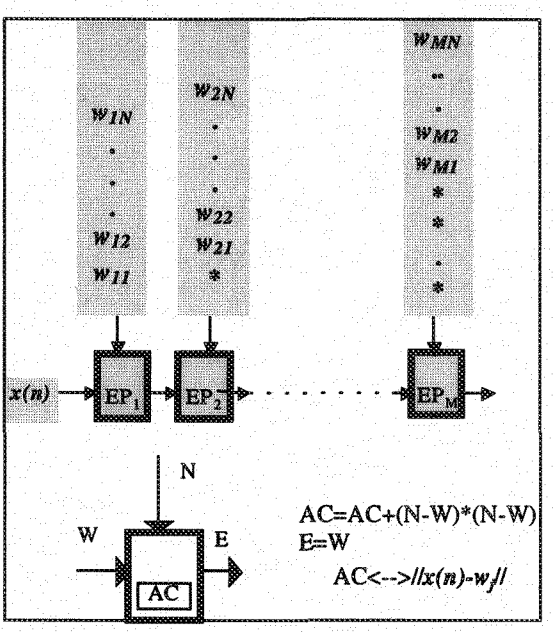

Fig. 8-3. Systolic algorithm for computing the distance.

When all the distances are computed, we can obtain the winning neuron by searching for the EP with the lower AC value.

The same scheme can be used with this purpose when performing a comparison operation such as the one described in Fig. 8-4.

We must include one communication between the last EP and the first one.

Each EP must include one comparator circuit.

When the M cycles are completed, the output of the last EP offers the minimum distance computed in the previous phase. By comparing this output with the accumulators of each EP, we obtain the winning neuron and begin the weight upgrading phase.

Before changing the weights, the η factor (with the same value for each EP) will be computed. The neighbor function γ will be initialized to 1 for the winning neuron and with lower values for the others neurons, according to the distance from this winning neuron. The γ values are computed in advance, and are stored in one memory array.

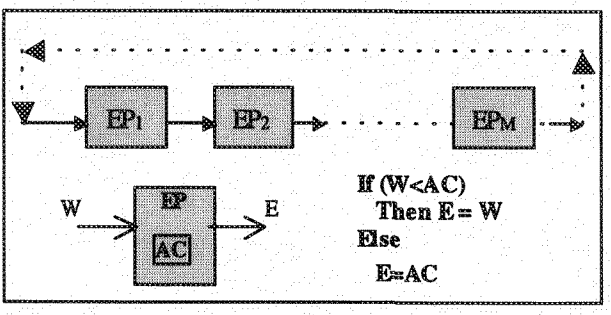

Fig. 8-4. Linear systolic scheme to determine the lower distance. (AC stores the distances previously computed).

In order to compute the weights according to Eq. (8-2), the following steps will be performed:

1. The x values stored in the first EP in the distance computation phase will be propagated to the other EPs.

2. When x_i arrives at the j element, the difference between the x_i value and the weight value w_{ji} is obtained and multiplied by the η and γ values of this EP. In this way, we obtain Δw_{ji} by adapting the weight of the j element.

Fig. 8-5 shows the scheme of this process. By following the previous systolic description, we can conclude:

- The systolic SOM implementation has M EPs.
- Each EP will be programmable to change functionality in the different phases.
- The memory for each EP must store one row of the weight matrix, and one FIFO is required to recycle the x data.
- The data are transmitted in a single direction between two neighbor EPs. To compute the winning neuron, we include one connection from the last neuron to the first.
- Each EP supports all the arithmetic processing capabilities (subtractions, comparisons, and accumulations).

The classification process includes resolution of the winning neuron, achieved by computation of the minimum distance.

In order to classify hyperspectral vectors provided by the DAIS sensors with 220 bands, the proposed neural network has one input neuron for each channel (220 neurons), and the number of output neurons is fixed at 17. The

systolic array includes one EP for each output neuron, and each EP has 220 memory cells to store the weights.

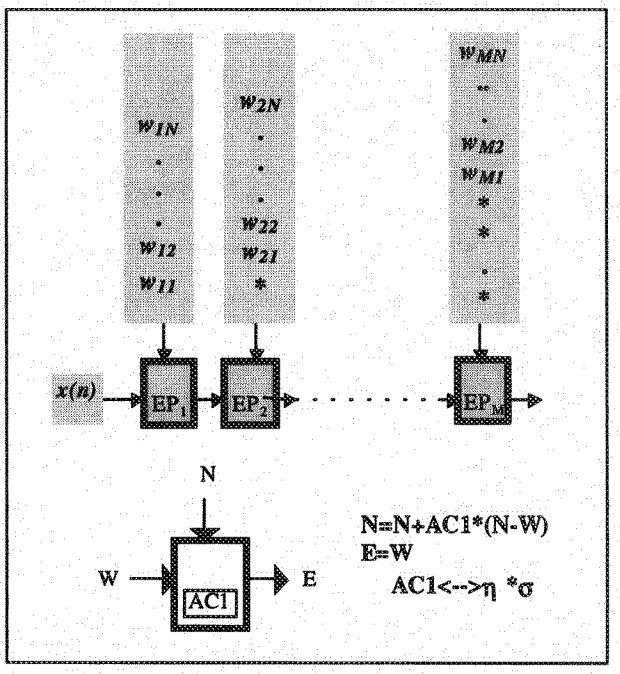

Fig. 8-5. Systolic scheme for the weight upgrading phase.

8.6. Conclusions

This chapter describes the design of one linear systolic array to implement one SOM neural network. The aim is to carry out unsupervised classification of hyperspectral images. The proposed architecture reduces classification time by using M EP's to perform parallel computations with N-element vectors. For the retrieving phase, the number of sequential operations is 4 MxN. The systolic array reduces this number to M+N. For the weight modification phase, the cycle number is reduced from 5NxM cycles for each iteration to N+M cycles for each iteration. This great reduction alleviates the processing of the image pixels by allowing real-time processing.

Acknowledgements

This chapter has been conceived under the funding project Aplicación de las imágenes hiperespectrales a la vigilancia de recursos naturales (TIC 2000-0739-C04-3), Ministerio de Educación y Ciencia (Spain).

We also wish to acknowledge the linguistic revision of the paper by Dr. Alejandro Curado Fuentes, from the Department of English at our Institution.

References

[1] A. F. H. Goetz, G. Vane, J. E. Solomon, and B. N. Rock. Imaging spectrometry for Earth remote sensing. *Science*, 211, pages 1147-1153, 1985.

[2] R. Green, M. Eastwood, T. Sarture, M. Aronson, J. Chippendale, J. A. Faust, B. Pavri, Ch. Chovit, M. Solis, M. Olah and O. Willians. Imaging Spectroscopy and the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS). *Remote Sensing of Environments*, 65, pages 227-248, 1998.

[3] S. Subramianian, N. Gat, A. Ratcliff and M. Eismann. Real-time Hyperspectral Data Compression Using Principal Components Transformation. *Summaries of the X Annual JPL Airborne Earth Science Workshop*, Jet Propulsion Laboratory, NASA, Pasadena ,California, pages 451-460, 2000.

[4] E. Merényi, The Challenges in Spectral Images: An Introduction and Review of ANN Approaches. *Proc. European Symposium on Artificial Neural Networks, ESANN'99,* Bruges, Belgium, pages 93-98, 1999.

[5] L. O. Jiménez, A. Morales-Morel and A. Creus. Classification of Hyperdimensional Data Based on Feature and Decision Fusion Approaches Using Projection porsuit, Mayority voting and Neural Networks. *IEEE Transactions on Geoscience and Remote Sensing*, 17(3), pages 1360-1366, 1999.

[6] T. Kohonen. *Self-Organizing Maps* (2nd. ed.), Springer Series in Information Science, 1997.

[7] J. Bruske and E. Merényi. Estimating the intrinsic dimensionality of Hyperspectral images. *Proc. European Symposium on Artificial Neural Network, ESANN'99,* Bruges, Belgium, 21-23, pages 105-110, April, 1999.

[8] P. L. Aguilar, P. Martínez, R.M. Pérez. Abundance extractions from AVIRIS image based on Self-Organizing neural network. *AVIRIS'2000, Earth Science and Applications Workshop,* Jet Propulsion Laboratory, NASA, Pasadena, California.

[9] P. L. Aguilar. *Cuantificación de firmas hiperespectrales usando mapas autoorganizativos,* Tesis Doctoral, Departamento de Informática, Escuela Politécnica, Cáceres, 2000.

[10] M. Zarghan. *Computer Architecture*, New Jersey, Prentice Hall, 1996

[11] R. M. Pérez, P. L. Aguilar, P. Bachiller and P. Martínez, Neural Network quantifier for solving the mixture problem and its implementation by systolic array. *Microelectronics Journal*, 30(1), pages 77-82, 1999.

[12] R. M. Pérez, P. Martínez, P. L. Aguilar y Linaje M., Arquitectura Sistólica para una Red SOM que Clasifica Imágenes Hiperespectrales, *XII Jornadas de Paralelismo*, Granada, Spain, pages 351-355, 2000.