Cellular Neural Networks Learning using Genetic Algorithm

Eduardo Gómez Ramírez, Ferran Mazzanti* & Xavier Vilasis Cardona*
Laboratorio de Investigación y Desarrollo de Tecnología Avanzada (LIDETEA)
UNIVERSIDAD LA SALLE
E-mail: egr@ci.ulsa.mx

*Departament d'Electrónica, Enginyeria i Arquitectura La Salle, U. Ramon Llull,
Pg. Bonanova 8, 08022 Barcelona, España
E-mail: mazzanti@salleURL.edu xvilasis@salleURL.edu

Recibido: Julio de 2002. Aceptado: Agosto de 200

ABSTRACT

In this paper an alternative to CNN learning using Genetic Algorithm is presented. The results show that it is possible to find different solutions for the cloning templates that fulfill the same objective condition. Different examples for image processing show the result of the proposed algorithm.

Keyword: Cellular neural network, learning, genetic algorithm, cloning templates, image processing.

RESUMEN

En este trabajo se propone una alternativa para el aprendizaje de Redes Neuronales Celulares utilizando algoritmo genético. Con las simulaciones realizadas se observó que es posible encontrar diferentes combinaciones de matrices de pesos con las que se obtienen los mismos resultados. Se muestran diferentes ejemplos para el procesamiento de imágenes para representar el comportamiento del algoritmo.

Palabras clave: Redes neuronales celulares, aprendizaje, algoritmo genético, plantilla donadora, procesamiento de imagen.

INTRODUCTION

Cellular Neural Networks (CNNs)[1] are arrays of dynamical artificial neurons (cells) locally interconnected. This essential characteristic has made possible the hardware implementation of large networks on a single VLSI chip[2] and optical computers.[3] The particular distribution of neurons in two–dimensional grids has turned CNNs into a particularly suitable tool for image processing.

Determining the correct set of weights for CNNs is traditionally a difficult task. This is, essentially, because gradient descent methods get rapidly stuck into local minima.[4] One finally resorts either to template the library, or sophisti-

[1] O. Chua & L. Yang: "Cellular Neural Networks: Theory", *IEEE Trans. Circuits Syst.,* vol. 35, pp. 1257-1272, 1988.
O. Chua & L. Yang: "Cellular Neural Networks: Applications", *IEEE Trans. Circuits Syst.,* vol. 35, pp.1273-1290, 1988.

[2] Liñan, L., Espejo, S., Dominguez-Castro, R., Rodriguez-Vazquez, A. ACE4K: an analog VO 64x64 visual microprocessor chip with 7-bit analog accuracy. Int. *J. of Circ. Theory and Appl.* en prensa, 2002.
Kananen, A., Paasio, A. Laiho, M., Halonen, K. CNN applications from the hardware point of view: video sequence segmentation. *Int. J. of Circ. Theory and Appl.* en prensa, 2002.

[3] Mayol W, Gómez E.: "2D Sparse Distributed Memory-Optical Neural Network for Pattern Recognition". *IEEE International Conference on Neural Networks,* Orlando, Florida, 28 de junio a 2 de julio, 1994

[4] M. Balsi, "Hardware Supervised Learning for Cellular an Hopfield Neural Networks", *Proc. of World Congress on Neural Networks,* San Diego, Calif, 4 al 9 de junio, pp. III, 451-456, 1994.

cated elaborations requiring a deep insight of the CNN behavior.[5] Among global optimization methods for learning, Genetic Algorithms reveal to be a popular choice[6]. However, authors usually show a genetic algorithm to find only one of the matrixes. The authors Doan, Halgamuge *et al*[7] solely show experiments for specific cases and possibly the examples presented only work with these constraints that, of course, reduce the space solution.

In this work a genetic algorithm approach is developed to compute all the parameters of a Cellular Neural Network (weights and current). The results show that it is possible to find many different matrixes that fulfill the objective condition. Different examples of image processing are presented to illustrate the procedure developed.

The structure of the paper is the following: Section 2 describes the theory of Cellular Neural Networks, Section 3 the genetic algorithm proposed and in Section 4 the way the algorithm was applied to find the optimal parameters for image processing.

CELLULAR NEURAL NETWORKS

As it was defined on its origins, a CNN is an identical array of non-linear dynamic circuits. This array could be made of different types depending on the way they are connected or the class of the neighborhood.

A. Mathematical Model

The simplified mathematical model that defines this array[8] is the following:

$$\frac{dx^c}{dt} = -x^c(t) + \sum_{d\in N_r(c)} a_d^c y^d(t) + \sum_{d\in N_r(c)} b_d^c u^d + i^c$$

(Eq.1)

$$y^c(t) = \frac{1}{2}\left(\left|x^c(t)+1\right| - \left|x^c(t)-1\right|\right)$$

(Eq.2)

where x^c represents the state of the cell c, y^c the output and u^c its input. The state of each cell is controlled by the inputs and outputs of the adjacent cells inside the r-neighborhood $N_r(c)$. The outputs are feedback and multiplied by a^c_d and the inputs are multiplied by control parameters b^c_d. This is the reason that a^c_d is called retro and b^c_d is called control. The value i^c is constant and is used for adjusting the threshold [9]. These coefficients are invariants to translations and it would be called cloning template.

The output y_c is obtained applying the equation (2). This equation limits the value of the output on the range between [-1,1], in the same way the activation function does in the classical ANN models (Fig. 1).

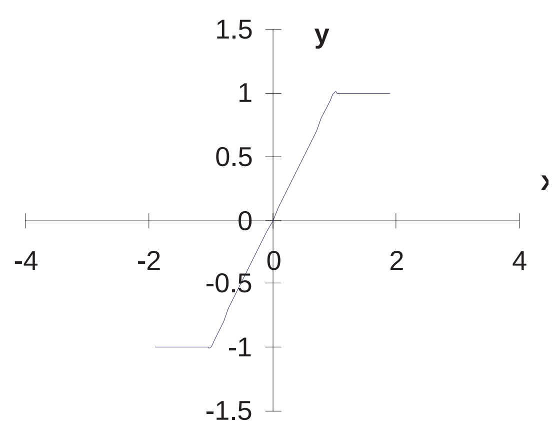

Figure 1. Output function

5 Zarandy, A., The Art of CNN Template Design. *Int. J. of Circ. Th. Appl.,* pp. 5-23, 1999.

6 Kozek, T., Roska, T., Chua, L.O. Genetic Algorithm for CNN Template Learning. *IEEE Trans. Circ. Syst.,* pp. 392-402, 1993.
P. López, M. Balsi, D.L. Vilario, D. Cabello, Design and Training of Multilayer Discrete Time Cellular Neural Networks for Antipersonnel Mine Detection using Genetic Algorithms, *Proc. of Sixth IEEE International Workshop on Cellular Neural Networks and their Applications,* Catania, Italia, 23 al 25 de mayo, 2000.
Taraglio S. & Zanela A., Cellular Neural Networks: age-netic algorithm for parameters optimization in artificial vision applications, CNNA-96., *Proceedings, Fourth IEEE International Workshop,* pp. 315-320, 24 al 26 de junio, Sevilla, España, 1996.

7 Doan, M. D., Halgamuge, S., Glesner M. & Braunsforth, Application of Fuzzy, GA and Hybrid Methods to CNN Template Learning, CNNA-96, *Proceedings, Fourth IEEE International Workshop*, pp. 327-332, 24 al 26 de junio, Sevilla, España, 1996.

8 Harrer H. & Nossek J.: "Discrete-Time Cellular Neural Networks". *Cellular Neural Networks.* Editado por T. Roska & J. Vandewalle. John Wiley & Sons, 1993.

9 This coefficient is equivalent to the input bias on other artificial neural networks.

B. Discrete Mathematical Model

The mathematical model illustrated before could be represented by the following expression:

$$\frac{dx^c}{dt} = -x^c(t) + k^c \qquad \text{(Eq.3)}$$

where:

$$k^c = \sum_{d \in N_r(c)} a_d^c y^d(t) + \sum_{d \in N_r(c)} b_d^c u^d + i^c$$

Because the final state is obtained evaluating x(∞), the model could be simplified as:

$$x^c(k) = \sum_{d \in N_r(c)} a_d^c y^d(k) + \sum_{d \in N_r(c)} b_d^c u^d + i^c \qquad \text{(Eq. 4)}$$

Some authors also considered a modification to the output (eq. 2), as follows [4]:

$$y^c(k) = f(x^c(k-1))$$

$$y^c(k) = \begin{cases} 1 & si\ \ x^c(k-1) > 0 \\ -1 & si\ \ x^c(k-1) < 0 \end{cases} \qquad \text{(Eq. 5)}$$

However, the simplification made (eq. 4) could be used with the function of the eq. 2.[10] [1]. Equation (4) can be represented using the convolution operator like:

$$X_k = A*Y_k + B*U_k + I \qquad \text{(Eq. 6)}$$

where matrix A and B correspond to the rotation of 180º of the corresponding coefficients a^c_d and b^c_d . The previous step is not necessary if the matrix has the property of $A=A^T$.

The cloning template, in some cases, is a set of different combinations that fulfill the same conditions, e.g., there are different matrixes that can solve the same problem. This can be a problem with the use of gradient error techniques, as in these cases, it is not possible to find an optimal solution. One alternative is the use of evolutionary computation techniques that can be applied to find the different solutions or alternatives to the same problem.

The next section describes the methodology of Genetic Algorithm used in this work.

GENETIC ALGORITHM

Genetic Algorithm (GA) is a knowledge model inspired on some of the evolution mechanisms observed in nature. GA usually follows the next cycle [11]: (Fig. 2)

- Generation of an initial population in a random way.
- Evaluation of the fitness or some objective function of every individual that belongs to the population.
- Creation of a new population by the execution of operations, like crossover and mutation, over individuals whose fitness or profit value have been measured.
- Elimination of the former population and iteration using the new one until the termination criteria is fulfilled or it has reached a certain number of generations.

In this work the crossover or sexual recombination, the mutation and other special process that we call 'add parents'[12] and 'add random parents' are used. Next Sections describe these processes.

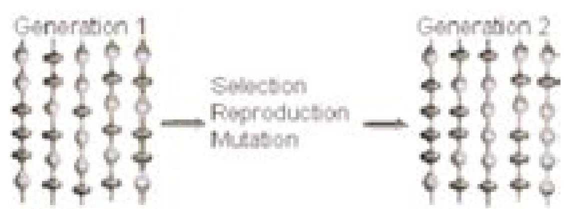

Figure 2. Operators of Genetic Algorithm

[10] M. Alencastre-Miranda, A. Flores-Méndez & E. Gómez Ramírez: "Comparación entre Métodos Clásicos y Redes neuronales celulares para el análisis de imágenes". *XXXVIII Congreso Nacional de Física.* Zacatecas, Zacatecas, México, 16 al 20 de octubre, 1995.

[11] Alander J. T., *An Indexed Bibliography of Genetic Algorithms*: Years 1957, 1993, 1994, Art of CAD Ltd. Bedner I., *Genetic Algorithms and Genetic Programming at Stanford,* Stanford Bookstore, 1997.

[12] E. Gómez-Ramírez, A. Poznyak, A. González Yunes & M. Avila-Alvarez. Adaptive Architecture of Polynomial Artificial Neural Network to Forecast Nonlinear Time Series. *CEC99 Special Session on Time Series Prediction.* Hotel Mayflower, Washington D.C., EUA, 6 al 9 de julio, 1999.

A. Crossover

The crossover operator is characterized by the combination of the genetic material of the individuals that are selected in function of the good performance (objective function).

To explain the multipoint crossover for each fixed number $g=1,..,n_g$, where n_g is the number of total generations, let introduce the matrix F_g which is the set of parents of a given population. This matrix is Boolean of dimension, F_g: $n_p x n_b$, where n_p is the number of parents of the population at the generation g and n_b is the size of every array (chromosomes). Let $C(F_g,n_t)$ be the crossover operator which can be defined as the combination between the information of the parents set considering the number of intervals n^t of each individual and the number of sons n_s such that:

$$n_s = n_p^{\,n_t}$$

(Eq. 7)

Then $C(F_g,n_t):n_p x n_b \rightarrow n_s x n_b$. To show how the crossover operator can be applied the following example is introduced. Let F_g has n_p=2 and n_t=2. This means that the array (the information of one father) is divided in 3 sections and every section is determined with a_i and b_i respectively for $i=1,..,n_t$. It's important to appoint that with this operator the parents F_g of the population g are included in the result of the crossover:

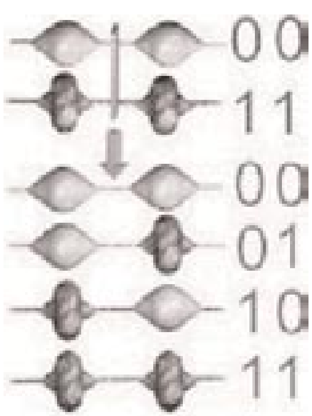

Figure 3. Multipoint Crossover Operator

$$F_g = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix}, M_1 = \begin{bmatrix} a & b \end{bmatrix} M_2 = \begin{bmatrix} c & d \end{bmatrix}$$

Example:

$$\Rightarrow C(F_g) = \begin{bmatrix} [a \quad b] \leftarrow \\ [a \quad d] \\ [c \quad b] \\ [c \quad d] \leftarrow \end{bmatrix}$$

Note that with this operator the parents F_g of the population g are included in the result of the crossover.

For the problem of CNN learning every section of one father can be represented by one column or one row of one matrix of the cloning template (matrix $a^c{}_d$ and $b^c{}_d$).

B. Mutation

The mutation operator just changes some bits that were selected in a random way from a fixed probability factor P_m; in other words, we just vary the components of some *genes*. This operator is extremely important, as it assures the maintenance of the diversity inside the population, which is basic for the evolution. This operator $M: n_s x n_b \rightarrow n_s x n_b$ changes with probability P_m a specific population in the following way:

$$M(F_{ij}, P_m) = \begin{cases} \overline{F_{ij}} & r(\omega) \leq P_m \\ F_{ij} & r(\omega) > P_m \end{cases}$$

(Eq. 8)

C. Add Parents Mutated

In this part the parents F_g are mutated and added to the result of the crossover process, then the population A_g at the generation g can be obtained like:

$$A_g = \begin{bmatrix} C(F_g) \\ M(F_g, P_m) \end{bmatrix}$$

(Eq. 9)

This step and the previous one ensure the convergence of the algorithm, because every generation at least has the best individual obtained in the process. With this step the original and mutated parents are included in the generation g and with some probability the individuals tend to the optimal one.

D. Selection Process

The Selection Process S_g computes the objective function O_g that represents a specific criterion to maximize or minimize and selects the best n_p individuals of A_g as:

$$S_g(A_g, n_p) = \min{}^{n_p} O_g(A_g)$$

(Eq. 10)

Then, the parents of the next generation can be calculated by:

$$F_{g+1} = S_g(A_g, n_p)$$

(Eq. 11)

Summarizing, the Genetic Algorithm can be describe with the following steps:

1. For the initial condition $g=0$ select the A_0 in a random way, such that $A_0 : n_s x n_b$
2. Compute $F_1 = S_0(A_0)$
3. Obtain A_g
4. Calculate S_g
5. Return to step 3 until the maximum number of generations is reached or one of the individuals of S_g obtains the minimal desired value of O_g

E. Add Random Parents

To avoid local minimal a new scheme is introduced and it is called 'add random parents'. If the best individual of one generation is the same than the previous one, a new random individual is included like parent of the next generation. This step increases the population because when crossover is applied with the new random parent, the number of sons increases. This step is tantamount to have a major mutation probability and to search in a new generation points of the solution space.

To apply the previous theory of GA, every individual is the corresponding matrix A (a^c_d) and B (b^c_d). The crossover is computed using every column of the matrix like previous explanations and the mutation is applied to every component of the matrix. In the current case only mutations are used.

APPLICATION IN IMAGE PROCESSING

The procedure used to train the network was:

1. To generate the input patterns using a known matrix in image processing to find the different kind of lines (horizontal, vertical, diagonal) in different images.
2. To train the network using the methodology explained above using GA.
3. Test the results using a new pattern.

The results were repeated to obtain different templates for the same problem. The parameters used are: 4,000 individuals in the initial population and 5 parents. The change in the values for every coefficient was evaluated using an increment of 0.125 to reduce the space solution.

Example 1 Horizontal Line Detector

Figures 4 and 5 show the input and output patterns used. This pattern was obtained using a well known matrix:

$$M = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

The parameters obtained are:

First Simulation:

$$A = \begin{bmatrix} -0.125 & 0 & 0 \\ 0 & 2.5 & 0.125 \\ -0.75 & -1.375 & 0.125 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.125 & -2 & 0 \\ 0.375 & 2.75 & 0 \\ 0.375 & -0.875 & -0.25 \end{bmatrix}$$

I=-2.875

Second Simulation

$$A = \begin{bmatrix} 4.5 & -2.25 & -0.5 \\ -1.625 & 0.75 & 3.375 \\ 2.125 & -0.875 & -0.875 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.75 & -7.125 & -0.625 \\ 2.625 & -0.875 & 2.25 \\ -3.125 & -0.875 & 2.375 \end{bmatrix}$$

I=0.625

Example 2 Vertical Line Detector

Figures 6 and 7 show the input and output patterns used. These patterns were obtained using a well known matrix:

$$M = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

In this case we put constraints to select only the matrix B. The parameters obtained are:

First simulation

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1.75 & -0.75 \\ 0 & 0 & 0 \end{bmatrix}$$

I=0

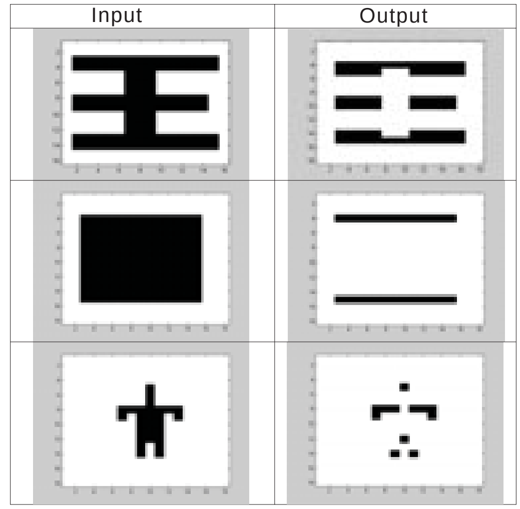

Figure 4. Input and Output patterns for CNN Learning

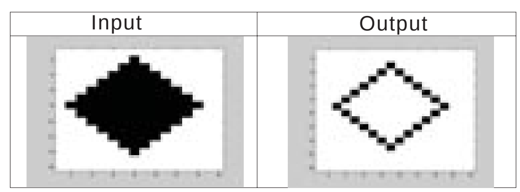

Figure 5. Input and Output Patterns for CNN Test

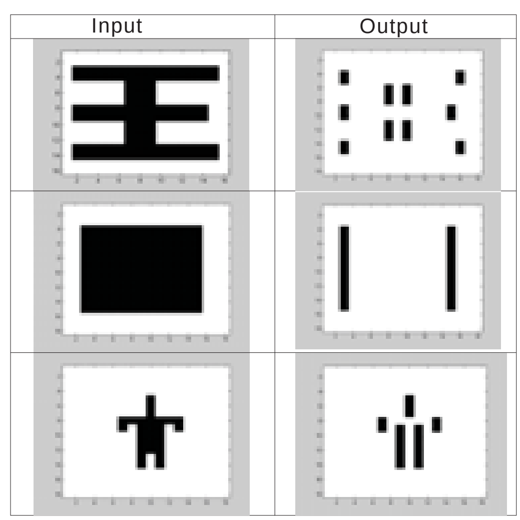

Figure 6. Input and Output patterns for CNN Learning

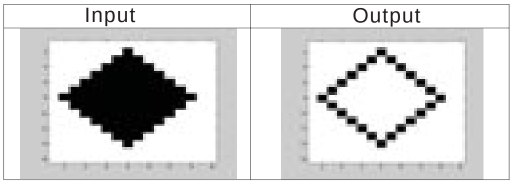

Figure 7. Input and Output Patterns for CNN Test

CONCLUSIONS

The learning of CNN is not a simple procedure, mainly due to the problem of computing the gradient of the estimation error. Genetic Algorithm is one alternative to solve it and it is possible to obtain different solutions for the same problem. In this work the use of 'add random parent' avoids reaching local minimal and improves the convergence of the algorithm. This is very important when the space solution includes all the elements of the cloning template.

REFERENCES

1. Chua, O. & L. Yang: "Cellular Neural Networks: Theory", *IEEE Trans. Circuits Syst.*, vol. 35, pp. 1257-1272, 1988.
2. Chua,O & L. Yang: "Cellular Neural Networks: Applications", *IEEE Trans. Circuits Syst.,* vol. 35, pp. 1273–1290, 1988.
3. Liñan, L., Espejo, S., Dominguez–Castro, R., Rodriguez–Vazquez, A. ACE4K: an analog VO 64x64 visual microprocessor chip with 7–bit analog accuracy. *Int. J. of Circ. Theory and Appl.* in Press, 2002.
4. Kananen, A., Paasio, A. Laiho, M., Halonen, K. CNN applications from the hardware point of view: video sequence segmentation. *Int. J. of Circ. Theory and Appl.,* en prensa, 2002.
5. Mayol W, Gómez E.: "2D Sparse Distributed Memory–Optical Neural Network for Pattern Recognition". *IEEE International Conference on Neural Networks,* Orlando, Florida, 28 de junio a 2 de julio, 1994.
6. Balsi, M. "Hardware Supervised Learning for Cellular and Hopfield Neural Networks", *Proc. of World Congress on Neural Networks,* San Diego, Calif, 4 al 9 de junio, pp. III, 451-456, 1994.
7. Zarandy, A., The Art of CNN Template Design. *Int. J. of Circ. Th. Appl.,* pp. 5–23, 1999.
8. Kozek, T., Roska, T., Chua, L.O. Genetic Algorithm for CNN Template Learning. *IEEE Trans. Circ. Syst.,* pp. 392–402, 1993.
9. López, P. M. Balsi, D.L. Vilario, D. Cabello, Design and Training of Multilayer Discrete Time Cellular Neural Networks for Antipersonnel Mine Detection using Genetic Algorithms, *Proc. of Sixth IEEE International Workshop on Cellular Neural Networks and their Applications,* Catania, Italia, 23 al 25 de mayo, 2000.
10. Taraglio S. & Zanela A., Cellular Neural Networks: a genetic algorithm for parameters optimization in artificial vision applications, CNNA–96., *Proceedings, Fourth IEEE International Workshop,* pp. 315–320, 24 al 26 de junio, Sevilla, España, 1996.
11. Doan, M. D., Halgamuge, S., Glesner M. & Braunsforth, Application of Fuzzy, GA and Hybrid Methods to CNN Template Learning, CNNA–96, *Proceedings, Fourth IEEE International Workshop,* pp. 327–332, 24 al 26 de junio, Sevilla, España, 1996.
12. Harrer H. & Nossek J.: "Discrete–Time Cellular Neural Networks". *Cellular Neural Networks.* Editado por T. Roska & J. Vandewalle. John Wiley & Sons, 1993.
13. Alencastre–Miranda, M. A. Flores–Méndez & E. Gómez Ramírez: "Comparación entre Métodos Clásicos y Redes neuronales celulares para el análisis de imágenes". *XXXVIII Congreso Nacional de Física.* Zacatecas, Zacatecas, México, 16 al 20 de octubre, 1995.
14. Alander J. T., An Indexed Bibliography of Genetic Algorithms: Years 1957, 1993, 1994, Art of CAD Ltd. Bedner I., *Genetic Algorithms and Genetic Programming at Stanford,* Stanford Bookstore, 1997.
15. Gómez–Ramírez, E., A. Poznyak, A. González Yunes & M. Avila–Alvarez. Adaptive Architecture of Polynomial Artificial Neural Network to Forecast Nonlinear Time Series. *CEC99 Special Session on Time Series Prediction.* Hotel Mayflower, Washington D.C., EUA, 6 al 9 de julio, 1999.