IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-5728, p-ISSN: 2319-765X. Volume 11, Issue 4 Ver. V (Jul - Aug. 2015), PP 62-67
www.iosrjournals.org

Numerical approach to the CNN Based Hole-Filler Template Design Using Leapfrog Method

S. Sekar[1], M. Vijayarakavan[2]
[1]*(Department of Mathematics, Government Arts College (Autonomous), Salem – 636 007, India)*
[2]*(Department of Mathematics, V.M.K.V. Engineering College, Salem – 636 308, Tamil Nadu, India)*

Abstract : *The Cellular Neural Network (CNN) is an artificial neural network of the nearest neighbour interaction type. It has been used for image processing, pattern recognition, moving object detection, signal processing, augmented reality and etc. The cellular neural network CMOS array was implemented by Anguita et al [1 - 5] and Dalla Betta et al [6]. The design of a cellular neural network template is an important problem, and has received wide attention [7 - 9]. Based on the dynamic analysis of a cellular neural network, this paper presents, a design method for the template of the hole-filler used to improve the performance of the handwritten character recognition using Leapfrog method. This is done by analyzing the features of the hole-filler template and the dynamic process of CNN and by using single-term Haar wavelet series method (STHW) and Leapfrog methods to obtain a set of inequalities satisfying its output characteristics as well as the parameter range of the hole-filler template. Some simulation results and comparisons are also presented.*

Keywords – *Cellular neural network, Hole-filler template, Haar wavelts, Single-Term Haar wavelts, Single-term Haar wavelet series, Leapfrog Method.*

I. INTRODUCTION

Cellular Neural Networks are analog, time continuous, nonlinear dynamic systems and formally belongs to the class of recurrent neural networks. Since their introduction in 1988 by Chua and Yang[10], they have been the subjects of intense research. Initial applications include image processing, signal processing, pattern recognition and solving partial differential equations etc. The cellular neural network is an artificial neural network of the nearest neighbor interaction type.

A similar hierarchical and cellular organization is revealed in language, which could be considered as an “image of our brain”. Basic cells (characters in the written language) collaborate in an emergent manner to form a word, which are then again combined to form a phrase and so on providing a structured reflection of the outer world. In a similar manner, cellular models can be provided for networks of similar electronic circuits, networks of computers and so on.

Recapitulating, universal characteristics of our world can be captured in the emergent dynamics of a cellular model where a population of similar cells exchanges information locally with the cells in their neighborhood. Emergent computation depends on how the information is exchanged and processed in the neighborhood of a cell.

Based on the dynamic analysis of a cellular neural network, this paper presents a design method for the template of the hole-filler used to improve the performance for the handwritten character recognition using Leapfrog method. By analyzing the features of the hole-filler template and the dynamic process of CNN, In this paper, describes the dynamic behavior of CNN in section 2, Hole-filler template design ideas in Section 3, Numerical Integration algorithms and its description is shown in Section 4, and simulation results in Section 5.

II. DYNAMIC ANALYSIS OF CNN

The dynamic equation of cell $C(i, j)$ in an M x N cellular neural network is given by Chua *et al.* [10].

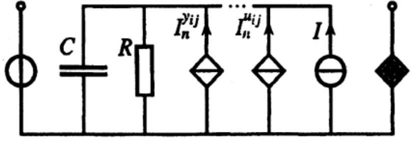


Fig. 1 CNN cell

$$C\frac{dx_{ij}(t)}{dt}=-\frac{1}{R_x}x_{ij}(t)+\sum_{C(k,l)\in N_r(i,j)}A(i,j;k,l)y_{kl}(t)+\sum_{C(k,l)\in N_r(i,j)}B(i,j;k,l)u_{kl}+I \qquad (1)$$

$$y_{ij}(t)=\tfrac{1}{2}\Big[\big|x_{ij}(t)+1\big|-\big|x_{ij}(t)\big|-1\Big],\ 1\le i\le M,\ 1\le j\le n \qquad (2)$$

where x_{ij}, y_{ij} and u_{ij} are the state voltage, output voltage and input voltage respectively and they are functions of time *t*. R_x is a linear resistance, C is a linear capacitor, and $A(i,j;k,l)$and $B(i,j;k,l)$ are the transconductances of the output and input voltages of $C(k,l)$ with respect to $C(i,j)$ called the cloning templates of CNN. $N_r(i,j)$ denotes the r^{th} -neighbour of $C(i,j)$ and I is an independent current source. From equation (2) one can see that the output voltage is nonlinear. We can rewrite the cell equation (1) as follows:

$$C\frac{dx_{ij}(t)}{dt}=-f\left[x_{ij}(t)\right]+g(t)$$

where

$$f\left[x_{ij}(t)\right]=\frac{1}{R_x}x_{ij}(t),$$

$$g(t)=\sum_{\substack{C(k,l)\in N_r(i,j)\\ C(k,l)\neq C(i,j)}}A(i,j;k,l)y_{kl}(t)+\sum_{C(k,l)}B(i,j;k,l)u_{kl}+I$$

III. HOLE-FILLER TEMPLATE DESIGN

The Hole-Filler is a cellular neural network, which fills up all the holes and remains unaltered outside the holes in a bipolar image. Let $R_x=1$, $C=1$ and let +1 stand for the black pixel and –1 for the white one. We shall discuss the images having holes enclosed by the black pixels, when the bipolar image is input with $U=\{u_{ij}\}$into CNN. The initial state values are set as $x_{ij}(0)=1$. From the equation (2) the output values are $y_{ij}(0)=1, 1\le i\le M, 1\le j\le N$

Suppose that the template *A* and *B* and the independent current source *I* are given as

$$A=\begin{bmatrix}0 & a & 0\\ a & b & a\\ 0 & a & 0\end{bmatrix},\quad a>0,\quad b>0,\quad B=\begin{bmatrix}0 & 0 & 0\\ 0 & 4 & 0\\ 0 & 0 & 0\end{bmatrix},\quad I=-1 \qquad (3)$$

where the template parameters *a* and *b* are to be determined. In order to make the outer edge cells become the inner ones, normally auxiliary cells are added along the outer boundary of the image, and their state values are set to zeros by circuit realization, resulting in the zero output values. The state equation (1) can be rewritten as

$$\frac{dx_{ij}(t)}{dt}=-x_{ij}(t)+\sum_{C(k,l)\in N_r(i,j)}A(i,j;k,l)y_{ij}(t)+4u_{ij}(t)-I \qquad (4)$$

For the cell $C(i,j)$, we call the cells $C(i+1,j), C(i-1,j), C(i,j+1)$ and $C(i,j-1)$ to be the non-diagonal cells. Here, several cases are to be considered

Case 1: The input value $u_{ij}=+1$ for cell $C(i,j)$, signaling the black pixel. Because the initial state value of the cell $C(i,j)$ has been set to 1, $x_{ij}(0)=1$, and from equation (2) its initial output value is also $y_{ij}(0)=1$. According to the hole-filler demands, its eventual output should be $y_{ij}(\infty)=1$. To obtain this result we set

$$\frac{dx_{ij}(t)}{dt}\ge 0 \qquad (5)$$

Substituting this input $u_{ij}=1$ and equation (3) into equation (4), we obtain

$$\frac{dx_{ij}(t)}{dt}=-x_{ij}(t)+a\left[y_{(i-1)j}(t)+y_{(i+1)j}(t)+y_{i(j-1)}(t)+y_{i(j+1)}(t)\right]+by_{ij}(t)+3 \qquad (6)$$

Combining equations (5) and (6) and considering the minimum value of $x_{ij}(t)=1$ this case yields

$$a\left[y_{(i-1)j}(t)+y_{(i+1)j}(t)+y_{i(j-1)}(t)+y_{i(j+1)}(t)\right]+by_{ij}(t)+2\ \ge\ 0 \qquad (7)$$

To facilitate our discussion, two sub cases are distinguished.

***Sub case* 1**: The cell $C(i, j)$ is inside the holes. Since $x_{ij}(0)=1$, from equation (2) its initial output value $y_{ij}(0)=1$. Considering equations (5) and (2), $y_{ij}(t) \geq 1$. According to the hole-filler demands, its initial output of non-diagonal black pixels should not be changed inside the holes. The weights of a and b are equal to +4 and +1, respectively. Since $A(i, j; i, j) > {}^{1}\!/_{R_x}$, the parameter b is found to be $b > 1$, or $4a + b + 2 \geq 0$ b is $b > 1$.

Sub case 2: The cell $C(i, j)$ is outside the holes. To satisfy equation (7), we need to check only the minimum value on the left-hand side of equation (7). This is true when there are four non-diagonal white pixels around the cell $C(i, j)$, where the weight of a *in* equation (7) is –4. Since $y_{ij}(t) \geq 1$, the weight of b is equal to 1. Combining this with $b > 1$ gives $-4a + b + 2 \geq 0$, b is $b > 1$.

Case 2: The input value of cell $C(i, j)$ is $u_{ij} = 1$, signaling the white pixel. Substituting this input value in equation (4) gives

$$\frac{dx_{ij}(t)}{dt} = -x_{ij}(t) + a\left[y_{(i-1)j}(t) + y_{(i+1)j}(t) + y_{i(j-1)}(t) + y_{i(j+1)}(t)\right] + by_{ij}(t) - 5 \qquad (8)$$

Sub case 1: The cell $C(i, j)$ is inside the holes. Since $x_{ij}(0)=1$, from equation (2) its initial output value is $y_{ij}(0)=1$. According to the hole-filler demands, the holes should be filled by the black pixels, whereas its initial black pixels remain unaltered:

$$\frac{dx_{ij}(t)}{dt} \geq 0 \qquad (9)$$

Combining equations (8) and (9) and considering $x_{ij}(t) \geq 1$ yields

$$a\left[y_{(i-1)j}(t) + y_{(i+1)j}(t) + y_{i(j-1)}(t) + y_{i(j+1)}(t)\right] + by_{ij}(t) - 6 \geq 0$$

where we use the minimum value of $x_{ij}(t)$ in equation (8). Since the cell is inside the holes, its initial output of non-diagonal black pixels remain unchanged. The weight of *a and b* are equal to +4 and +1, respectively. Combining this with $b > 1$ gives

$$4a + b - 6 \geq 0, \text{ b is } b > 1. \qquad (10)$$

Sub case 2: The cell $C(i, j)$ is outside the holes. Since $x_{ij}(0)=1$, from equation (2) its initial output value is $y_{ij}(0)=1$. According to the hole-filler demands, the final output of this cell should be white, and in this case $y_{ij}(\infty) \leq -1$.

$$\frac{dx_{ij}(t)}{dt} < 0 \qquad (11)$$

Combining equations (8) and (11) and considering $x_{ij}(t) \leq 1$ we get

$$a\left[y_{(i-1)j}(t) + y_{(i+1)j}(t) + y_{i(j-1)}(t) + y_{i(j+1)}(t)\right] + by_{ij}(t) - 6 < 0 \qquad (12)$$

where we use the maximum value of $x_{ij}(t)$ in equation (8)

Initially $y_{ij}(0)=1$. How can the output of cell $C(i, j)$ be changed to -1?. Where does this change begin?. First we consider the situation where the change begins from the inside of the bipolar image. If the maximum value on the left-hand side in equation (12) is less than zero, equation (12) holds. Inside the image and outside the holes, the maximum weights of *a* and *b are* +4 and +1, respectively. This case was described by equation (10). In fact, the change of the output of the cell $C(i, j)$ is like a wave propagating from the edges to the inside of the image and it is verified from the simulated result. Therefore, we should first consider the edge cell $C(i, j)$, $i = 1$or M, $j = 1$or *N*. For this the maximum weight of *a* in equation (12) is +3, which is also the maximum weight of *a* outside the holes. The maximum weight of *b* is +1, occurring at the initial time:

$$3a + b - 6 < 0,$$
$$b > 1.$$

Combining Cases 1 and 2, we obtain

$$3a + b - 6 < 0,$$
$$4a + b - 6 \geq 0,$$
$$-4a + b + 2 \geq 0$$

IV. NUMERICAL EXPERIMENTS FOR LEAPFROG METHOD

The CNN is described by a system of nonlinear differential equations. Therefore, it is necessary to discretize the differential equation for performing simulations. For computational purpose, a normalized time differential equations describing CNN is used by Nossek et al [11].

$$f'(x(\pi\tau)) := \frac{dx_{ij}(\pi\tau)}{dt} = -x_{ij}(\pi\tau) + \sum_{C(k,l)\in N_r(i,j)} A(i,j;k,l)y_{kl}(\pi\tau) + \sum_{C(k,l)\in N_r(i,j)} B(i,j;k,l)u_{kl} + I$$

$$y_{ij}(\pi\tau) = \frac{1}{2}\left(\left| x_{ij}(\pi\tau)+1\right| - \left| x_{ij}(\pi\tau)-1\right| \right)$$

Where τ is the normalised time. Well-established numerical integration techniques are used for implementing hole-filler template design. These methods can be derived using the definition of the definite integral

$$x_{ij}((n+1)\tau) - x_{ij}(\pi\tau) = \int_{\tau_n}^{\tau_{n+1}} f'(x(\pi\tau))d(\pi\tau)$$

The most familiar and elementary method for approximating solutions of an initial value problem is Euler's Method. Euler's Method approximates the derivative in the form of $y' = f(t,y),\ y(t_0) = y_0,\ y \in R^d$ by a finite difference quotient $y'(t) \approx (y(t+h) - y(t))/h$. We shall usually discretize the independent variable in equal increments:

$t_{n+1} = t_n + h,\ n = 0,1,\ldots t_0$.

Henceforth we focus on the scalar case, N = 1. Rearranging the difference quotient gives us the corresponding approximate values of the dependent variable:

$y_{n+1} = y_n + hf(t_n, y_n),\ n = 0,1,\ldots t_0$

To obtain the leapfrog method, we discretize t_n as in $t_{n+1} = t_n + h,\ n = 0,1,\ldots t_0$, but we double the time interval, *h*, and write the midpoint approximation $y(t+h) - y(t) \approx hy'\left(t + \frac{h}{2}\right)$ in the form

$y'(t+h) \approx (y(t+2h) - y(t))/h$

and then discretize it as follows:

$y_{n+1} = y_{n-1} + 2hf(t_n, y_n),\ n = 0,1,\ldots t_0$

The leapfrog method is a linear *m* = 2-step method, with $a_0 = 0, a_1 = 1, b_{-1} = -1, b_0 = 2$ and $b_1 = 0$. It uses slopes evaluated at odd values of *n* to advance the values at points at even values of *n*, and vice versa, reminiscent of the children's game of the same name. For the same reason, there are multiple solutions of the leapfrog method with the same initial value $y = y_0$. This situation suggests a potential instability present in multistep methods, which must be addressed when we analyze them—two values, y_0 and y_1, are required to initialize solutions of $y_{n+1} = y_{n-1} + 2hf(t_n, y_n),\ n = 0,1,\ldots t_0$ uniquely, but the analytical problem $y' = f(t,y),\ y(t_0) = y_0,\ y \in R^d$ only provides one. Also for this reason, one-step methods are used to initialize multistep methods.

V. SIMULATION RESULTS

This Hole-filler template has been simulated using Pentium Core 2 Duo Machine with 3.0 Ghz. speed using STHW and Leapfrog methods of hand written image.

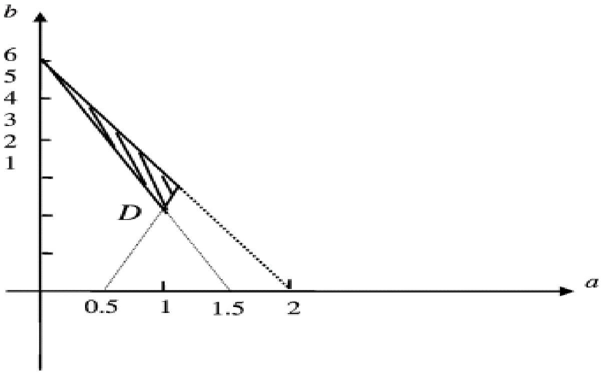
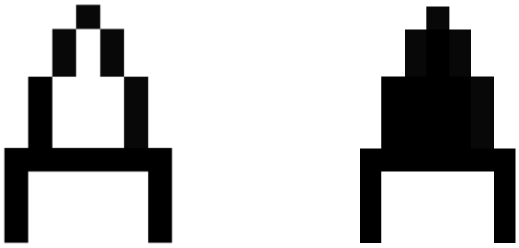


Fig. 2 The parameter range of template



(a) (b)

Fig. 3 Image before and after hole-filling

Step size (h)	STHW Method		Leapfrog Method	
	Settling Time (T)	Integration Time (Ts)	Settling Time (T)	Integration Time (Ts)
0.5	6.5	2.5	5.8	2.4
0.6	15.5	12.7	11.4	12.5
0.7	32.5	28.3	30.0	27.2
0.8	35.0	30.7	32.4	29.6
0.9	36.8	32.6	34.2	31.6
1.0	37.9	33.6	36.0	32.8
1.5	44.8	36.8	41.1	36.0
2.0	47.4	43.2	46.2	42.8
2.5	50.6	45.6	48.3	44.7
3.0	53.5	49.3	52.3	49.2

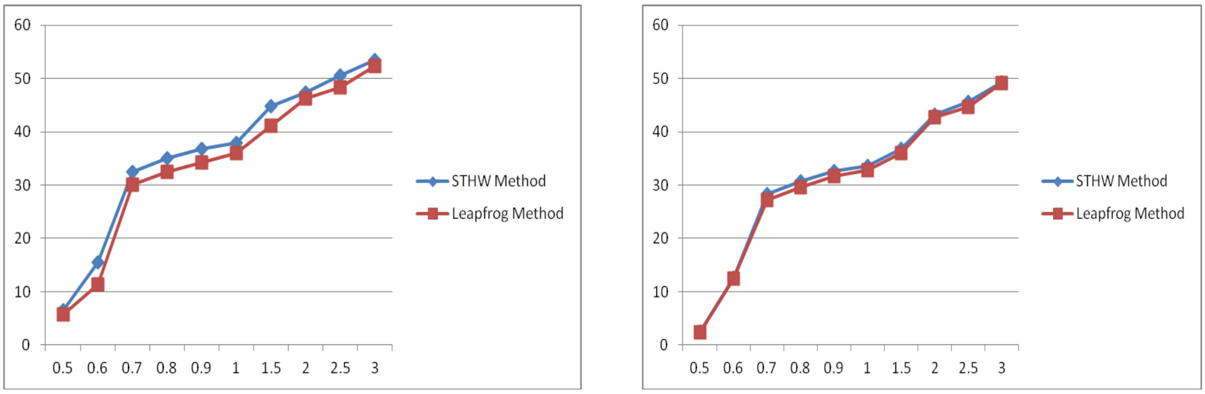
Table 1 Simulated results of hole-filler template design

The Settling time *T* and integration time *Ts* is obtained with various step sizes is to be displayed below in the Table 1. The settling time *Ts* describes the time from start of integration until the last cell leaves the interval **[-1.0, 1.0]** which is based on certain limit (e.g., |dx / dt| < 0.01). The simulation shows the desired output for every cell. We use +1 and -1 to indicate the black and white pixels, respectively. We remark that the selected parameters a and *b* of the template are restricted to the shaded area of Fig. 2. during the simulation.

Example

The template A, whose parameters are within the shaded area of Figure-2, and template B and *I* are given as follows:

$$A=\begin{bmatrix} 0 & 1.0 & 0 \\ 1.0 & 3 & 1.0 \\ 0 & 1.0 & 0 \end{bmatrix}, \qquad B=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I=-1.0$$



(a) (b)

Fig. 4 Comparison Results (a) Setting Time (b) Integration Time

Using the simulation program, the input image is shown in Fig. 3(a) and the output image in Fig. 3(b). The obtained results are represented in the form of a graph in Fig. 4. (a) and Fig. 4. (b). From the graph we find that Leapfrog method yields less settling time and integration time compared to STHW method.

VI. CONCLUSION

It is shown that the cellular neural network based hole-filler template could be designed from its dynamic behavior using STHW and Leapfrog methods, and also the template for other cellular neural network can similarly be designed. The hole is filled and the outside image remains the same. The templates of the cellular neural network are not unique and this is important in its implementation.

Acknowledgements

The authors gratefully acknowledge the Dr. A. Murugesan, Assistant Professor, Department of Mathematics, Government Arts College (Autonomous), Salem - 636 007, for encouragement and support. The authors also heartfelt thank to Dr. S. Mehar Banu, Assistant Professor, Department of Mathematics, Government Arts College for Women (Autonomous), Salem - 636 008, Tamil Nadu, India, for her kind help and suggestions.

REFERENCES

[1] M. Anguita, F. J. Pelayo, E. Ros, D. Palomar and A. Prieto, "VLSI implementations of CNNs for image processing and vision tasks: single and multiple chip approaches", *IEEE International Workshop on Cellular Neural Networks and their Applications,* 1996, pp. 479 - 484.

[2] M. Anguita, F. J. Pelayo, F. J. Fernandez and A. Prieto, "A low-power CMOS implementation of programmable CNN's with embedded photosensors", *IEEE Transactions on Circuits Systems I: Fundamental Theory and Applications,* Vol. 44, No.2, 1997, pp. 149 - 153.

[3] M. Anguita, F. J. Pelayo, E. Ros, D. Palomar and A. Prieto, "Focal-plane and multiple chip VLSI approaches to CNNs", *Analog Integrated Circuits and Signal Processing,* Vol. 15, No. 3, 1998, pp. 263 - 275.

[4] M. Anguita, F. J. Pelayo, I. Rojas and A. Prieto, "Area efficient implementations of fixed template CNN's", *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications,* Vol. 45, No. 9, 1997, pp. 968 - 973.

[5] M. Anguita, F. J. Fernandez, A. F. Diaz, A. Canas and F. J. Pelayo, "Parameter configurations for hole extraction in cellular neural networks", *Analog Integrated Circuits and Signal Processing,* Vol. 32, No. 2, 2002, pp.149 - 155.

[6] G. F. Dalla Betta, S. Graffi, M. Kovacs and G. Masetti, "CMOS implementation of an analogy programmed cellular neural network", *IEEE Transactions on Circuits and Sysems - part–II*, Vol. 40, No. 3, 1993, pp. 206 - 214.

[7] M. Boroushaki, M. B. Ghofrani and C. Lucas, "Simulation of nuclear reactor core kinetics using multilayer 3-D cellular neural networks", *IEEE Transactions on Nuclear Science*, Vol. 52, No. 3, 2005, pp.719 - 728.

[8] L. O. Chua and P. Thiran, "An analytic method for designing simple cellular neural networks", *IEEE Transactions on Circuits and Systems*, Vol. 38, No. 11, 1991, pp. 1332 - 1341.

[9] F. Fajfar, T. Bratkovic, T. Tuma and J. Puhan, "A rigorous design method for binary cellular neural networks", *International Journal of Circuit Theory and Applications,* Vol. 26, No. 4, 1998, pp. 365 - 373.

[10] L. O. Chua and L. Yang, "Cellular neural networks: theory", *IEEE Transactions on Circuits and Systems*, Vol. 35, No. 10, 1988, pp. 1257 - 1272.

[11] J. A. Nossek, G. Seiler, T. Roska and L. O. Chua, "Cellular neural networks: theory and circuit design", *International Journal of Circuit Theory and Applications*, Vol. 20, 1992, pp. 533 – 553.