OPEN ACCESS
World J of Engineering and Pure and Applied Sci. 2012;**2:20** ISSN 2249-0582
***Babatunde et al.**, 2012. Simulation of a Boundary Valued-Neuronal Model*

Original Article
Basic Science

On Numerical Simulation of a Boundary Valued-Neuronal Model

Oluleye Hezekiah BABATUNDE [1*], Olufisoye Caleb AKANBI [1], ,Adekunle ELUDIRE [2], Oluwaseun Gbenga FADARE [2], Olumide Gabriel EGBEDOKUN [3], Olabisi Babatope ALUKO [4]

ABSTRACT [ENGLISH/ANGLAIS]

This work presents the analysis and numerical solution of a boundary-valued Neuronal Model- *Cellular Neural Networks (CNN)* on a finite lattice with various boundary conditions imposed. The influence of these boundary conditions (*Fixed (or Dirichlet*), *Zero-flux (or Neumann*), *Periodic (or toroidal)* on the CNN Dynamics for adaptive contour detection and certain system theoritic property, are investigated. The results and analysis in this investigation have significant implications for circuit design in cellular neural networks and different applications of CNN such as image processing, solution of PDEs and other real life problems.

Keywords: Neuronal model, cellular neural networks), boundary conditions, simulation

RÉSUMÉ [FRANÇAIS/FRENCH]

Ce travail présente la solution d'analyse et numériques d'une frontière à valeur de modèle cellulaire neuronal-Neural Networks (CNN) sur un réseau fini avec les conditions aux limites imposées différents. L'influence de ces conditions aux limites (fixe (ou de Dirichlet), de flux nul (ou de Neumann), périodique (ou torique) sur la dynamique de CNN pour la détection des contours adaptatifs et certains biens theoritic système, sont étudiées. Les résultats et l'analyse de cette enquête ont des implications importantes pour la conception de circuits dans les réseaux cellulaires de neurones et des applications différentes de CNN comme le traitement de l'image, la solution d'EDP et d'autres problèmes de la vie réelle.

Mots-clés: Modèle neuronal, des réseaux cellulaires de neurones), les conditions aux limites, simulation

Affiliations:

[1] Department of Information and Communication Technology, Osun State University Osogbo, Osun State, NIGERIA

[2] Department of Computer Science, Joseph Ayo Babalola University Ikeji-Arakeji, Osun State, NIGERIA

[3] Department of Computer Science, The Polytechnic Ibadan, Oyo State, NIGERIA

[4] Department of Mathematics and Statistics, Osun State Polytechnic Iree, Osun State, NIGERIA

Address for Correspondence/ Adresse pour la Correspondance: hezecomp@yahoo.com

Accepted/Accepté: December, 2011

Citation: Babatunde OH, Akanbi OC, Eludire A, Fadare OG, Egbedokun OG, Aluko OB. On Numerical Simulation of a Boundary Valued-Neuronal Model. World Journal of Engineering and Pure and Applied Sciences 2012;2:20-5.

INTRODUCTION

Cellular neural networks (CNN) are special class of neural networks, with the unique feature that allow communication between neighboring units only. CNN has many real life applications ranging from image processing [1], image morphism, solution of partial differential equations [2], reduction of non-visual problems to geometric maps, modeling of mammalian retina and other sensory-motor organs [3]. CNN was invented in 1988 by Chua and Yang at the University of California, Berkeley (Department of Electrical Engineering and Computer Sciences) [4, 5].

A Cellular Neural Network (CNN) is an array of cells or coupled networks with local connections only. There are several topological layout for CNN Cells. The most popular is the two-dimensional CNNs organized in an eight-neighbor rectangular grid. A cell in a CNN is a dynamical system consisting of a state ,input and an output, and it interacts directly only with the cells within its *radius of neighborhood r*: The r is commonly fixed to be = 1(Fig. 1). The state of each cell and its output depends only on the input and the output of its neighbor cells, and the initial state of the network. With different entries in A,B,I (i.e., its interaction weights), a CNN can present a large number of dynamics, as proven by Ercsey et al [5].

Image processing is probably the most widespread application of CNN Models. Of recent, it has been proved that they can be also used for simulations in fluid dynamics [2] and statistical physics [5] Efficient Performance of this model is largely dependent on the imposed boundary conditions. This paper presents the numerical simulation of CNN Model alongside with the influence of different boundary conditions (*Fixed (or*

Dirichlet), *Zero-flux (or Neumann)*, *Periodic (or toroidal)*) on the CNN Dynamics

Figure 1: This figure shows two-dimensional CNN with radius of neighborhood r=1': the red cell has nine neighbors (the eight blue cells and itself) [6]

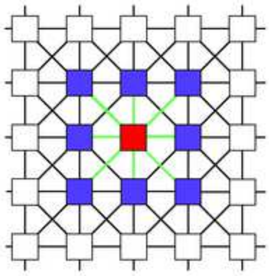


Figure 2: This figure shows standard nonlinearity for the output equation [6]

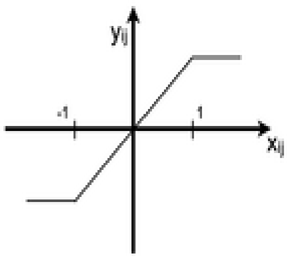
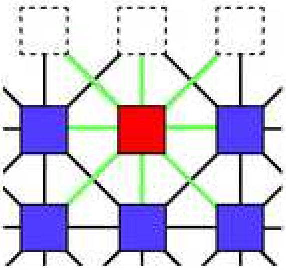


Figure 3: This figure shows local connections of an edge cell. Observe that three of its neighbors are boundary cells (dashed)



Circuitry and Modeling of Cellular Neural Networks

The CNN dynamics is described by a system of nonlinear differential equations. Using the simplest first-order cell dynamics and linear interactions, the general state equation of a cell in position *(i,j)* is as follows:

$$\frac{dx_{ij}(t)}{dt} = -x_{ij}(t) + \sum_{(k,l)\in N(i,j)} A(i,j:k,l)\cdot y_{kl}(t) + \sum_{(k,l)\in N(i,j)} B(i,j:k,l)\cdot u_{kl}(t) + z(i,j:k,l) \qquad (2.1)$$

The variables u_{ij}, x_{ij}, and y_{ij} depicts the input, the state, and the output of the cell in position *(i,j)*, respectively; The fundamental cell is denoted by the indices *k* and *l* belonging to the neighborhood $N(i,j)$ of the cell in position *(i,j)*. All variables are continuous. The triple *{A,B,z}* is the determinant of the CNN and it contains the weights of the neural/nonlinear network. The triple is called cloning template which determines the functionality of the CNN.In space invariant CNN, the dynamic behavior of the network depends only on a few parameters; for instance, for a two-dimensional CNN with a *radius of neighborhood r=1*, *A* and *B* are 4 x 4 matrices, while z is a scalar; therefore, in total just 33 real numbers determines the CNN dynamics [8].

The expression for the output $y_{i,j}$(diagramatized in Figure 2), is

$$y_{ij}(t) = f(x_{ij}(t)) = \tfrac{1}{2}\left(\left|x_{ij}(t)+1\right| - \left|x_{ij}(t)-1\right|\right) \qquad \ldots\ldots (2.2)$$

It is note-worthy that the constraints $|x_{ij}(0)|\le 1$, $1\le i\le M$; $1\le j\le N$, and $|u_{ij}(0)|\le 1$, $1\le i\le M$; $1\le j\le N$ are fundamentally imposed on the internal state of neuron (i,j) and input of neuron (i,j)

The original Chua Model was derived by circuitry modeling and some evolution law (Figure 4):

$$C\frac{dx_{ij}(t)}{dt} = -\frac{1}{R_x}x_{ij}(t) + \sum_{C(k,l)\in N_r(i,j)} A(i,j:k,l)\cdot y_{kl}(t) + \sum_{C(k,l)\in N_r(i,j)} B(i,j:k,l)\cdot u_{kl}(t) + I(i,j:k,l) \qquad (2.3)$$

Where $x_{ij}(t) \Rightarrow$ internal state of neuron (i,j), $y_{ij} \Rightarrow$ output of neuron (i,j) , $u_{ij} \Rightarrow$ input of neuron (i,j) , $A(i,j:k,l) \Rightarrow$ output feedback parameter, $B(i,j:k,l) \Rightarrow$ input control parameter, $C(i,j) \Rightarrow$ neuron (i,j), $N_r(i,j) \Rightarrow$

$$\{(k,l) : \max\{|k-i|, |l-j|\} \le r, 1\le k\le M, 1\le l\le N\}$$

$\Rightarrow$ The -neighborhood of neuron (i,j),

$I \Rightarrow$ Independent voltage source, $C \Rightarrow$ Linear Capacitor, $R_x \Rightarrow$ Linear Resistor

In this conventional CNN, there are usually positive self-feedbacks $A(i,j:k,l) \geq 0$

Certain conditions of convergence must be met by the tripple {A,B,z} which will make the CNN converge to a steady

Figure 4: This figure shows typical circuit of CNN [7]

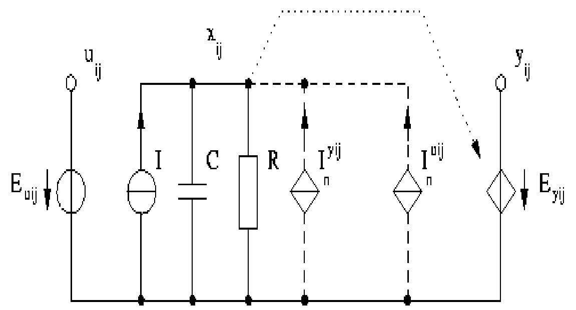


Figure 5: This figure shows lock diagram of one cell [7]

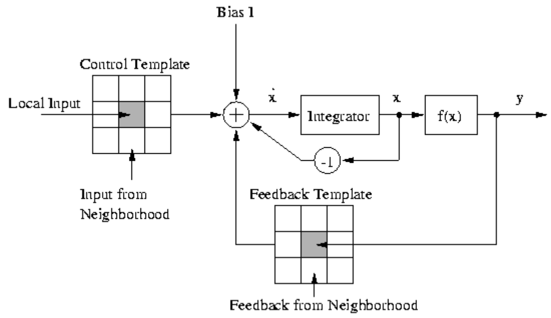
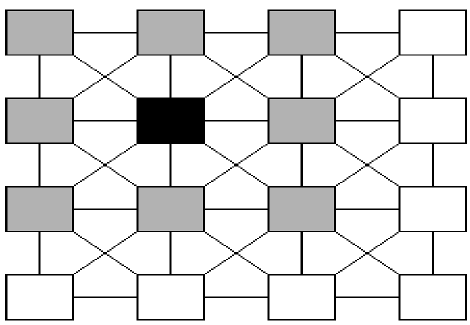


Figure 6: This figure shows CNN ij – position [7]



Numerical Simulation of Cellular Neural Networks

The equation (2.0) is represented in programming language statement/code segment as

```
D       =       -X[i][j] + A(i,j) + B(i,j)+ Z
x[ ]    =       {-1,0,1,-1,0,1,-1,0,1};
y[ ]    =       {-1,-1,-1,0,0,0,1,1,1};
double s =      0.0;
for     (int i=0;i<9;i++)
        s = s+ Y[xp+x[i]][yp+y[i]]*Atempl.v[y[i]+1][x[i]+1];
```

The state **X** of the CNN is derived by runge-kutta method

In the most general case, the final state of one cell can be described by the following equation:

$$x(t) = x(t_0) + \int_{t_0}^{t} \dot{x}(\tau)d\tau = x(t_0) + \int_{t_0}^{t} f(x(\tau))d\tau$$

and the integrating method for this could be any of Euler,RK,FEM etc. The RK method used is given as

$$\int_{t_n}^{t_{n+1}} f(x(t))dt = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Where

$$k_1 = \Delta t \cdot f(x(t_n))$$

$$k_2 = \Delta t \cdot f(x(t_n) + \frac{1}{2}k_1)$$

$$k_3 = \Delta t \cdot f(x(t_n) + \frac{1}{2}k_2)$$

$$k_4 = \Delta t \cdot f(x(t_n) + \frac{1}{2}k_3)$$

The Neumann boundary condition is the zero flux boundary condition. The states of the boundary sites are set equal to the states at the corresponding neighboring sites in CNN Namely, for $0 \leq i \leq k_1 + 1$, $0 \leq j \leq k_2 + 1$,

$$u_{k_1+1,j} = u_{k_1,j}, \quad u_{0,j} = u_{1,j}, \quad u_{i,k_2+1} = u_{i,k_2},$$

$$u_{i,0} = u_{i,1}$$ i.e

0	0	0	A	A	A
0	2	0	A	B	A
0	0	0	A	A	A

The periodic boundary condition identifies the first and the last rows (respectively columns) of the array Tk, thereby forming a torus. Namely, for $0 \leq i \leq k_1 + 1$, $0 \leq j \leq k_2 + 1$,

$$u_{1,j} = u_{k_1,j}, \quad u_{0,j} = u_{k_1-1,j}, \quad u_{2,j} = u_{k_1+1,j},$$

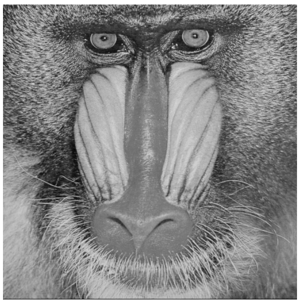
$$u_{i,1} = u_{i,k_2}, \quad u_{i,0} = u_{i,k_2-1}, \quad u_{i,2} = u_{i,k_2+1}.$$

The Dirichlet boundary condition means that certain boundary data (fixed constants)are prescribed on the boundary sites

INPUT DATA

The major idea in this work is to input a digital image in any format. The simulation was done in CNN Visual Mouse Platform. The embedded codes extract pixel information from the image in form of numerical data. These data are in turn used to generate different outputs shown below. The contents of A,B,z and their topology made the CNN performed differently during the simulation [3, 8].

Experimental Results



Original Image

Fixed Value Boundary Contour

Zero flux Boundary

Periodic Boundary Contour

Dynamic Properties and Stability

Let us consider the difference equation for equation (2.1) using the simple numerical Euler's algorithm

$$x_{ij}(t+1) = (1-\tfrac{\Delta t}{CR_x})x_{ij}(t) +$$

$$\tfrac{\Delta t}{C} \sum_{C(k,l)\in N(i,j)} A(i,j:k,l)\cdot y_{kl}(t) +$$

$$+ \tfrac{\Delta t}{C} \sum_{C(k,l)\in N_r(i,j)} B(i,j:k,l)\cdot u_{kl}(t) +$$

$$\tfrac{\Delta t}{C}\ z(i,j:k,l)\ m[y_{ij}(t)-z_0]$$

$$x_{ij}(t+1) = px_{ij}(t) + \sum_{C(k,l)\in N(i,j)} a(i,j:k,l)\cdot y_{kl}(t) +$$

$$\sum_{C(k,l)\in N_r(i,j)} b(i,j:k,l)\cdot u_{kl}(t) + z - m[y_{ij}(t)-z_0]$$

Where we have added a self-feedback for each neuron as indicated by the last term in

$$p = 1-\tfrac{\Delta t}{CR_x}, a(i,j;k,l) = A(i,j;k,l)\tfrac{\Delta t}{C},$$

$$b(i,j;k,l) = b(i,j;k,l)\tfrac{\Delta t}{C}, Z = z\tfrac{\Delta t}{C}$$

m = self-feedback connection weight , z_0 = A positive bias factor .

Positivity of this feedback implies stability while negativity implies chaos and oscillations [6]. Unstable dynamic behaviors will occur if the magnitude of the negative self-feedback is sufficiently large and we may change the chaotic system to transiently chaotic system by gradually removing the negative self-feedback. By setting the feedback to decay exponentially we have

$m(t+1) = (1-\phi)m(t)$ where ϕ, ($0 \le \phi \le 1$) is the damping factor of the time-dependent self-feed back $m(t)$

The resulting equation by combining (2.4) with the activation function and the self-feedback decaying function is the new model

$$x_{ij}(t+1) = mx_{ij}(t) + \sum_{C(k,l)\in N(i,j)} a'(i,j:k,l)\cdot y_{kl}(t) + \sum_{C(k,l)\in N_r(i,j)} b(i,j:k,l)\cdot u_{kl}(t) + z' \quad (2.5)$$

$$y_{ij}(t) = \tfrac{1}{2}(|x_{ij}(t)+1| - |x_{ij}(t)-1|) \quad (2.6)$$

$$m(t+1) = (1-\phi)m(t) \quad (2.7)$$

Where

$$a'(i,j:k,l) = a(i,j:k,l) - m(t) \quad (2.8)$$

$$z' = z + m(t)z_0 \quad (2.9)$$

To compare (2.5) with (2.1), is better to write (2.5)-(2.7) as

$$\tfrac{dx_{ij}(t)}{dt} = -m_{ij}(t)(y_{ij}(t)-z_0) + px_{ij}(t) + \sum_{(k,l)\in N(i,j)} A(i,j:k,l)\cdot y_{kl}(t) + \sum_{(k,l)\in N(i,j)} B(i,j:k,l)\cdot u_{kl}(t) + z(i,j:k,l) \quad (2.10)$$

$$y_{ij}(t) = \tfrac{1}{2}(|x_{ij}(t)+1| - |x_{ij}(t)-1|) \quad (2.11)$$

$$\frac{dm_{ij}(t)}{dt} = -\phi m_{ij}(t) \quad (2.12)$$

Stability of CNN

An appropriate Energy function according to Lipo et al. [9] to use for this TC-CNN is

$$E(t) = -\tfrac{1}{2}\sum_{(i,j)}\sum_{(k,l)} a'(i,j:k,l)\cdot y_{ij}(t)\cdot y_{kl}(t) - \sum_{(i,j)}\sum_{(k,l)} b(i,j:k,l)y_{ij}(t)\cdot u_{kl}(t) - \sum_{(i,j)} z' y_{ij}(t) + \tfrac{1-p}{2}\sum_{(i,j)} y_{ij}(t)^2 \quad (2.13)$$

The difference between two time steps in the energy function is

$$\Delta E = -\tfrac{1}{2}\sum_{(i,j)}\sum_{(k,l)} a'(i,j:k,l)\cdot \Delta y_{ij}(t)\cdot \Delta y_{kl}(t) - \sum_{(i,j)}\sum_{(k,l)} a'(i,j:k,l) y_{kl}(t)\cdot \Delta y_{ij}(t) - \sum_{(i,j)} \Delta y_{ij}(t)[\sum_{(k,l)} b(i,j;k,l)u_{kl} + z'] + \tfrac{1-p}{2}\sum_{(i,j)} \Delta y_{ij}(t)[y_{ij}(t+1) - y_{ij}(t)]. \quad (2.14)$$

From Cell Circuit Theory (2.14) can also be written as

$$\Delta E = -\tfrac{1}{2}\sum_{(i,j)}\sum_{(k,l)} a'(i,j:k,l)\cdot \Delta y_{ij}(t)\cdot \Delta y_{kl}(t) - \sum_{(i,j)} \Delta y_{ij}(t)[x_{ij}(t+1) - p x_{ij}(t)] + \tfrac{1-p}{2}\sum_{(i,j)} \Delta y_{ij}(t)[y_{ij}(t+1) - y_{ij}(t)].$$

From the output function (2.2), we have

$y_{ij}(t) = x_{ij}(t),$ when $|x_{ij}(t)| < 1$ and

$\Delta y_{ij}(t) = 0,$ when $|x_{ij}(t)| \geq 1$

Then

$$\Delta E = -\tfrac{1}{2}\sum_{|x_{ij}|<1}\sum_{|x_{kl}|<1} a'(i,j:k,l)\cdot \Delta y_{ij}(t)\cdot \Delta y_{kl}(t) - \sum_{|x_{ij}|<1} \Delta y_{ij}(t)[y_{ij}(t+1) - p y_{ij}(t)] + \tfrac{1-p}{2}\sum_{|x_{ij}|<1} \Delta y_{ij}(t)[y_{ij}(t+1) + y_{ij}(t)].$$

$$\Delta E = -\tfrac{1}{2}\sum_{|x_{ij}|<1}\sum_{|x_{kl}|<1} [a'(i,j:k,l) + (1+p)\delta_{ik}\delta_{jl}] * \Delta y_{ij}(t)\Delta y_{kl}(t).$$

Therefore, according to lyapunov theorem, if the matrix $[a'(i,j:k,l) + (1+p)\delta_{ik}\delta_{jl}]$ is positive definite, i. e $\Delta E \leq 0,$ then the network is stable .Hence, a sufficient stability condition for the TC-CNN is

$$a'(i,j:k,l) > -(1+p) = \tfrac{\Delta t}{CR_x} - 2 \quad (2.16)$$

where CR_x is the dynamic time constant of system (2.1) . Substituting for (2.8) in (2.16)

$$a(i,j:k,l) - m(t) > -(1+p) \quad (2.17)$$

If $p = 0.7,$ based on (2.16), the system is stable and converges to a fixed point when

$$a(i,j:k,l) - m(t) > -(1+p) = -1.7$$

CONCLUSION

We have investigated the effect of constant boundary conditions of CNN global dynamic behavior. The results shown above(having different pixel resolution) shows that the variation of Matrix A and B (Boundary Conditions of CNN) plays a major role in its performance. As a case study we have considered CNNs described by one-dimensional templates. System Theoretic property with respect to stability of the TC-CNN is also investigated. Boundary conditions are one of the structural features on which it seems to be of great interest to focus in order to obtain more knowledge about dynamical systems useful to medicine and biology.

REFERENCES

[1] Rudin LI, Osher S, Fatemi E. Nonlinear total variation based noise removal algorithms Multigrid Method for a Modified CDD Model for Image Inpainting. 875 Physica D 1992;60:259-68.

[2] Gilli M, Roska T, Chua Y, Leon O, Civalleri B,Pier P. CNN dynamics represents a broader class than PDEs. International Journal of Bifurcation and Chaos 2002;12(10):2051-68.

[3] Chen H, Dai M, Wu Y. Bifurcation and chaos in discretetime cellular neural networks. in Proc. 3rd IEEE Int. Workshop on Cellular Neural Netw. and Appl 1994;309–14.

[4] Hezekiah BO, Folorunso OT, Akinwale AT. A Cellular Neural Networks-Based Model for Edge

Detection. Journal of Information and Computing Science 2010;5:1-8.

[5] Ercsey R, Maria M, Néda Z, Roska T. Statistical Physics on Cellular Neural Network Computers. Physica D: Nonlinear Phenomena 2008;237(9):2051-68.

[6] Wikipedia. Cellular neural network. Accessed from: http://en.wikipedia.org/wiki/Cellular_neural_network . Last accessed: December 2011.

[7] Chua LO, Yang L. Cellular Neural Networks: theory and applications. IEEE Trans on Circuits and System 1988; 35(10):1257-72

[8] Aziz W, Lara T. Controllability, Applications and Numerical simulations of Cellular Neural Networks Electronic Journal of Differential Equation Conference 2005;13:1– 11.

[9] Lipo W, Wen L, Haixiang S, Jacek M, Zurada N. Cellular Neural Networks With Transient Chaos IEEE Transactions on Circuits and Systems—Iiexpress Briefs 2007;54(5):23-33.

ACKNOWLEDGEMENT / SOURCE OF SUPPORT

Nil

CONFLICT OF INTEREST

No conflict of interests was declared by authors