ON THE SIMULATION OF TIME DERIVATIVE CELLULAR NEURAL NETWORKS

Oguzhan Yavuz, S. Nergis Tural Polat and Vedat Tavsanoglu

Department of Electronics and Communications Engineering
Yildiz Technical University, Istanbul, Turkey
{ogyavuz, nergis, tavsanav}@yildiz.edu.tr

ABSTRACT

In this paper our previously proposed TDCNN simulation formulation is rewritten in vector-matrix form and thus a matrix condition for the empirical time constraint given in [1] is derived. The sinusoidal simulation results for band-pass filter example are presented.

1. INTRODUCTION

In our previous study, we introduced a new simulation method for time derivative cellular neural networks (TDCNN) with first derivatives [1]. The method in [1] uses forward Euler approximation for the derivatives on the left hand side, and backward Euler approximation for the derivatives on the right hand side of the TDCNN equation and computes the state of each cell by using convolution sums thus provides a great speed advantage.

In this paper, the formulation for the simulation of TDCNN used in [1] is given in vector-matrix form which enables the derivation of a matrix condition that formally proves the time constraint given empirically in [1]. We show that our TDCNN simulation method in [1] has the same solution as the forward Euler method iff the eigenvalues λ_i of $\hat{\mathbf{A}}_1$ are negative. Then we present the numerical simulation results of the method given in [1] and forward Euler method, and show that the results are consistent.

2. TIME-DERIVATIVE CELLULAR NEURAL NETWORKS (TDCNN)

Time-derivative CNN (TDCNN) [2] extends the original CNN description in [3] by adding derivative connections between cells. A time-derivative linear CNN is described by

$$\begin{aligned}\frac{dx(i,j,t)}{dt} &= \sum_{m,n=-r}^{r} A(m,n)\,x(i+m,j+n,t) \\ &+ \sum_{m,n=-r}^{r} B(m,n)\,u(i+m,j+n,t) \\ &+ \sum_{q\in 1..D}\left[\begin{array}{l}\sum_{m,n=-r}^{r} A_q(m,n)\dfrac{dx^q(i+m,j+n,t)}{dt^q} \\ + \sum_{m,n=-r}^{r} B_q(m,n)\dfrac{du^q(i+m,j+n,t)}{dt^q}\end{array}\right]\end{aligned} \tag{1}$$

The first two terms on the right hand side of (1) are the same as in the case of original CNN equation, A and B are feedback and feed-forward cloning templates, u is input and x denotes the state and output of the linear network. A_q and B_q are defined as qth derivative feedback and feedforward templates, respectively, and r denotes the neighborhood of the CNN. It has been shown that by adding first order derivatives of the outputs of the neighboring cells to the original CNN equation, bandpass spatiotemporal filters can be realized [2,4]. For these first derivative TDCNNs (1) becomes

$$\begin{aligned}\frac{dx(i,j,t)}{dt} &= \sum_{m,n=-r}^{r} A(m,n)\,x(i+m,j+n,t) \\ &+ \sum_{m,n=-r}^{r} B(m,n)\,u(i+m,j+n,t) \\ &+ \sum_{m,n=-r}^{r} A_1(m,n)\frac{dx(i+m,j+n,t)}{dt}\end{aligned} \tag{2}$$

3. ANALYSIS OF TDCNN SIMULATION METHODS

Equation (2) can be written in vector-matrix form by using one of the several packing schemes. Thus for a network size of MxN cells, MNx1 size vector-matrix differential equation of TDCNN is obtained as

$$\frac{d\mathbf{x}}{dt} = \hat{\mathbf{A}}\mathbf{x} + \hat{\mathbf{B}}\mathbf{u} + \hat{\mathbf{A}}_1\frac{d\mathbf{x}}{dt} \tag{3}$$

Here $\hat{\mathbf{A}}, \hat{\mathbf{A}}_1$ and $\hat{\mathbf{B}}$ are MNxMN matrices, $\mathbf{x}$ and $\mathbf{u}$ are MNx1 vectors that includes all the cell outputs and inputs respectively. We can rearrange (3) so that all the derivative terms are on the left hand side of the equation

$$\left(\mathbf{I}-\hat{\mathbf{A}}_1\right)\frac{d\mathbf{x}}{dt} = \hat{\mathbf{A}}\mathbf{x} + \hat{\mathbf{B}}\mathbf{u}\,. \tag{4}$$

where $\mathbf{I}$ denotes identity matrix of MNxMN size, which yields

$$\frac{d\mathbf{x}}{dt} = \left(\mathbf{I}-\hat{\mathbf{A}}_1\right)^{-1}\hat{\mathbf{A}}\mathbf{x} + \left(\mathbf{I}-\hat{\mathbf{A}}_1\right)^{-1}\hat{\mathbf{B}}\mathbf{u}\,. \tag{5}$$

Let us now apply Euler's forward approximation to (5):

$$\frac{\mathbf{x}(k+1)-\mathbf{x}(k)}{T_s} = \left(\mathbf{I}-\hat{\mathbf{A}}_1\right)^{-1}\hat{\mathbf{A}}\mathbf{x}(k) + \left(\mathbf{I}-\hat{\mathbf{A}}_1\right)^{-1}\hat{\mathbf{B}}\mathbf{u}(k) \tag{6}$$

Rearranging (6) yields

$$\mathbf{x}(k+1)=\mathbf{x}(k)+T_s\left(\mathbf{I}-\hat{\mathbf{A}}_1\right)^{-1}\left[\hat{\mathbf{A}}\mathbf{x}(k)+\hat{\mathbf{B}}\mathbf{u}(k)\right] \quad (7)$$

In [1] we have used forward Euler difference for the derivative on the left hand side and the backward Euler difference for the derivatives on the right hand side of (3), resulting in the difference equation:

$$\frac{\mathbf{x}(k+1)-\mathbf{x}(k)}{T_s}=\hat{\mathbf{A}}\mathbf{x}(k)+\hat{\mathbf{B}}\mathbf{u}(k)+\hat{\mathbf{A}}_1\left[\frac{\mathbf{x}(k)-\mathbf{x}(k-1)}{T_s}\right] \quad (8)$$

and

$$\mathbf{x}(k+1)=\left(\mathbf{I}+\hat{\mathbf{A}}_1\right)\mathbf{x}(k)+T_s\left[\hat{\mathbf{A}}\mathbf{x}(k)+\hat{\mathbf{B}}\mathbf{u}(k)\right]-\hat{\mathbf{A}}_1\mathbf{x}(k-1) \quad (9)$$

Equation (9) can be rearranged as

$$\begin{aligned}\mathbf{x}(k+1)&=\mathbf{x}(k)+T_s\left(\mathbf{I}-\hat{\mathbf{A}}_1\right)^{-1}\left[\hat{\mathbf{A}}\mathbf{x}(k)+\hat{\mathbf{B}}\mathbf{u}(k)\right]\\&+\left(\mathbf{I}-\hat{\mathbf{A}}_1\right)^{-1}\left(-\hat{\mathbf{A}}_1\right)\left[\mathbf{x}(k+1)-2\mathbf{x}(k)+\mathbf{x}(k-1)\right]\end{aligned} \quad .(10)$$

If the absolute values of eigenvalues of matrix $\left(\mathbf{I}-\hat{\mathbf{A}}_1\right)^{-1}\left(-\hat{\mathbf{A}}_1\right)$ are less than 1, the powers of this matrix approaches zero as the iteration continues. Consequently the last term in (10) decreases and we obtain the same equation as (7). In [1] we pointed out that the input image is held constant for at least 3 iterations for our method to give the same results as SIMULINK simulation. This condition corresponds to having very small values for $\left[\left(\mathbf{I}-\hat{\mathbf{A}}_1\right)^{-1}\left(-\hat{\mathbf{A}}_1\right)\right]^3$ after three iterations, thus obtaining

$$\mathbf{x}(k+1)=\mathbf{x}(k)+T_s\left(\mathbf{I}-\hat{\mathbf{A}}_1\right)^{-1}\left[\hat{\mathbf{A}}\mathbf{x}(k)+\hat{\mathbf{B}}\mathbf{u}(k)\right]. \quad (11)$$

In other words, the time constraint for the simulation of TDCNN given in [1] must be satisfied to ensure that the powers of the $\left(\mathbf{I}-\hat{\mathbf{A}}_1\right)^{-1}\left(-\hat{\mathbf{A}}_1\right)$ decrease.

Let us now examine the eigenvalues of $\left(\mathbf{I}-\hat{\mathbf{A}}_1\right)^{-1}\left(-\hat{\mathbf{A}}_1\right)$. First we decompose $\hat{\mathbf{A}}_1$ as

$$\hat{\mathbf{A}}_1=\mathbf{T}_1\Lambda_1\mathbf{T}_1^{-1} \quad (12)$$

where $\Lambda_1=diag(\lambda_i)$ and λ_i's are the eigenvalues of $\hat{\mathbf{A}}_1$. Now we can write

$$\begin{aligned}\left(\mathbf{I}-\hat{\mathbf{A}}_1\right)^{-1}\left(-\hat{\mathbf{A}}_1\right)&=\left(\mathbf{I}-\mathbf{T}_1\Lambda_1\mathbf{T}_1^{-1}\right)^{-1}\left(-\mathbf{T}_1\Lambda_1\mathbf{T}_1^{-1}\right)\\&=\mathbf{T}_1\left[\left(\mathbf{I}-\Lambda_1\right)^{-1}\left(-\Lambda_1\right)\right]\mathbf{T}_1^{-1}=\mathbf{T}_1\left[\left(\mathbf{I}-\Lambda_1^{-1}\right)^{-1}\right]\mathbf{T}_1^{-1}\end{aligned} \quad (13)$$

Which shows that the eigenvalues of $\left(\mathbf{I}-\hat{\mathbf{A}}_1\right)^{-1}\left(-\hat{\mathbf{A}}_1\right)$ are $\frac{-\lambda_i}{1-\lambda_i}$. If $\lambda_i<\frac{1}{2}$ then

$$\left|\frac{-\lambda_i}{1-\lambda_i}\right|<1 \quad (14)$$

Hence for $k\Rightarrow\infty$ $\left[\left(\mathbf{I}-\hat{\mathbf{A}}_1\right)^{-1}\left(-\hat{\mathbf{A}}_1\right)\right]^k\Rightarrow 0$.

4. SIMULATION RESULTS

In this section we present the simulation results for the three methods given above.

The input image is spatio-temporal, thus we have time varying frames of images.

For the simulations we must first evaluate the MNxMN size $\hat{\mathbf{A}},\hat{\mathbf{A}}_1$ and $\hat{\mathbf{B}}$ matrices. Since for a 20x20 TDCNN, the size of the matrices would be 400x400, a 4x4 size TDCNN example with the templates $A=\begin{bmatrix}0&1&0\\1&-4&1\\0&1&0\end{bmatrix}$, $B=\begin{bmatrix}0&0&0\\0&1&0\\0&0&0\end{bmatrix}$ and $A_1=\begin{bmatrix}0&0&0\\0&-1&0\\0&0&1\end{bmatrix}$ is given.

For this example we have:

$$\hat{\mathbf{A}}=\begin{bmatrix}
-4&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
1&-4&1&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&1&-4&1&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&1&-4&0&0&0&1&0&0&0&0&0&0&0&0\\
1&0&0&0&-4&1&0&0&1&0&0&0&0&0&0&0\\
0&1&0&0&1&-4&1&0&0&1&0&0&0&0&0&0\\
0&0&1&0&0&1&-4&1&0&0&1&0&0&0&0&0\\
0&0&0&1&0&0&1&-4&0&0&0&1&0&0&0&0\\
0&0&0&0&1&0&0&0&-4&1&0&0&1&0&0&0\\
0&0&0&0&0&1&0&0&1&-4&1&0&0&1&0&0\\
0&0&0&0&0&0&1&0&0&1&-4&1&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&1&-4&0&0&0&1\\
0&0&0&0&0&0&0&0&1&0&0&0&-4&1&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&1&-4&1&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&1&-4&1\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&-4
\end{bmatrix}$$

$$\hat{\mathbf{A}}_1 = \begin{bmatrix}
-1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1
\end{bmatrix}$$

$$\hat{\mathbf{B}} = \mathbf{I}$$

$$\mathbf{\Lambda}_1 = -\mathbf{I}$$

The eigenvalues of $\left[\left(\mathbf{I}-\hat{\mathbf{A}}_1\right)^{-1}\left(-\hat{\mathbf{A}}_1\right)\right]^3$ are given by

$$\left[\frac{-\lambda_i}{1-\lambda_i}\right]^3 = 0.125$$

We simulated the spatio-temporal bandpass filter TDCNN example given in [2]. Bandpass filter outputs are given for forward Euler method and the method in (8) in Fig. 1 and Fig. 2 respectively. As can be seen from the figures, the simulation results are consistent.

5. CONCLUSION

General 3D continuous-time discrete-space mixed-domain spatio-temporal filters can be realized by TDCNNs. In this paper it is proven that the method given in [1] and forward Euler method has the same solution under the condition that the absolute value of eigenvalues of $\left(\mathbf{I}-\hat{\mathbf{A}}_1\right)^{-1}\left(-\hat{\mathbf{A}}_1\right)$ are less than 1. The necessary and sufficient condition for this outcome is that the eigenvalues λ_i of $\hat{\mathbf{A}}_1$ should be negative. The sinusoidal simulation results of forward Euler method and proposed method in [1] are consistent with each other.

ACKNOWLEDGMENT

The authors would like to acknowledge Drs. Emmanuel M. Drakakis and Henry Man Den Ip for helpful discussions.

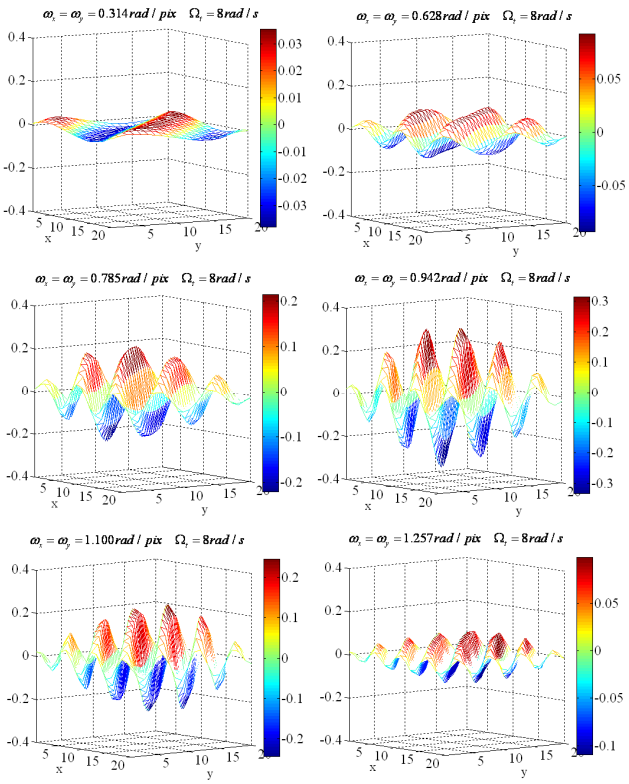

Figure 1. Bandpass filter simulation (for the template values given in [2]) of forward Euler method. The passband of the filter is around

$$\omega_x = \omega_y = 1 rad/pix, \Omega_t = 8 rad/s$$

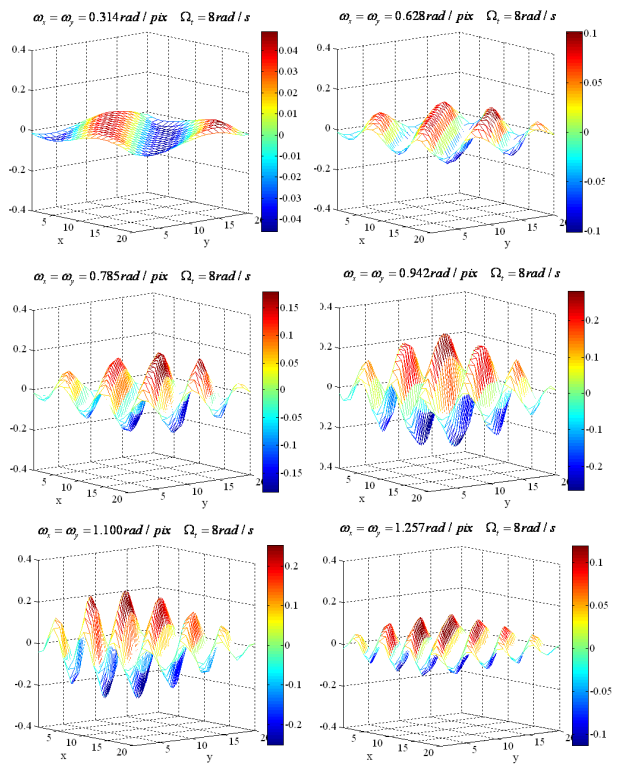

Figure 2. Bandpass filter simulation (for the template values given in [2]) of our previously proposed method [1]. The passband of the filter is around $\omega_x = \omega_y = 1 rad/pix, \Omega_t = 8 rad/s$

REFERENCES

[1] Tural Polat, N., and Tavsanoglu, V., “A new simulation method for time-derivative CNNs”, “17th European Signal Processing Conference (EUSIPCO’09)”, Glasgow, Scotland, pp. 288-292, 2009.

[2] H. M. D. Ip, E. M. Drakakis, and A. A. Bharath, “Towards analog vlsi arays for nonseparable 3-D spatiotemporal filtering,” in Proc. IEEE Int. Workshop Cellular Neural Networks and their Applications (CNNA’06), Istanbul,Turkey, 2006, pp. 1–6.

[3] L. O. Chua and L. Yang, “Cellular neural networks: Theory and applications”, IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 35, pp. 1257–1290, Oct. 1988

[4] H. M. D. Ip, E. M. Drakakis, and A. A. Bharath, “On analog networks and mixed-domain spatiotemporal frequency response,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 2, pp. 272–285, Feb. 2008.