Cellular Neural Network Computational Scheme for Efficient Implementation of the FDTD method.

Juan Manuel Mayor\textsuperscript{1} Luis Eduardo Tobon \textsuperscript{1} Zoltan Nagy\textsuperscript{2}
Eugenio Tamura\textsuperscript{1}

\textsuperscript{1}Pontificia Universidad Javeriana, Cali, Colombia
\textsuperscript{2}Pazmany Peter Catholic University, Hungary

2013 IEEE AP-S/URSI-USNC Symposium
Outline

1. CNN Basics
2. CNN and FDTD mapping
3. Parallelized Algorithm
4. Results
5. Conclusion
Cellular Neural Network Proposal

Cellular Neural Networks have been represented through state-of-the-art as a new computational parallel scheme. To solve some dynamics system (PDEs) [1] based on the inherit parallel properties.

**Figure:** Naive Scheme for training an ANN over hypothetical Maxwell PDE prediction in the case of specific EM problem.
Use parallel flexibility

We can also work not only with the layer parallelization, but with the complete parallelization per neuron that we called a cell due to its function variability in this context [2].

Figure: CNN structure taking the parallelizing properties for all the cell involved on it, in this case we have full connectivity per each cell.
Parallel Between ANN and CNN

Figure: Comparison between ANN and CNN structures per unit, in the Cell case we have the dynamic of linear one order RC - circuit model with independent current sources.
CNN Scheme Characteristics

ANN could diverge compared with CNN in terms of the activation function and variability of the Weights. But also in terms of the quantities of independant current sources in the scheme.

\[
\frac{x_0(t)}{R} + C\frac{dx_0}{dt} + I_{\text{bias}} = \sum_{k=0}^{M} i_k + \sum_{j=0}^{N} B_j x_0^j
\]  

(1)

Now if we fix this equation using the common Taylor forward differences we can arrive to

\[
x_0(n) = -x_0(n-1) \left( \frac{1}{RC} + 1 \right) + \sum_{k=0}^{M} A_k x_k(n-1) + \sum_{j=1}^{N} \sum_{i=0}^{M} B_j^i x_0(n-j, i)
\]  

(2)
In equation 2 we have all the dynamic behavior that is presented in each cell individually [3].

- $n$: This is the index of the discrete time analog to timestep unit.
- $x_0$: The state variable of every cell in the array (output voltage in the circuit scheme), it could be E or H in our specific case.
- $A_k$: is the neighborhood matrix that is related with the parameters that multiply the state variables
- $I_k$: This is the current dependent source k-index for the k-neighbor cell in the array, $(i_k = A_k x_k)$.
- $B_k$: is the matrix of initial and previous time values that affect $x_0$ or the state variable of the cell itself, It could be represented as a series of matrices in terms of time, only varying in space domain matrix.
- $I_{bias}$: Bias current analog with the bias parameter in ANN, this parameter is optional
3D FDTD-CNN mapping

Following the basic model of Maxwell PDE equation we have the following PDE discretized equations two for each pair of axis.

\[ \begin{align*}
H^{n+\frac{1}{2}} &= \left( \frac{2\mu - \Delta t\sigma_m}{2\mu + \Delta t\sigma_m} \right) H^{n-\frac{1}{2}} - \left( \frac{\Delta t}{(2\mu + \Delta t\sigma_m)} \right) \nabla \times E^n - \left( \frac{\Delta t}{(2\mu + \Delta t\sigma_m)} \right) M^n \\
E^{n+1} &= \left( \frac{2\varepsilon - \Delta t\sigma_e}{2\varepsilon + \Delta t\sigma_e} \right) E^n + \left( \frac{\Delta t}{(2\varepsilon + \Delta t\sigma_e)} \right) \nabla \times H^{n+\frac{1}{2}} - \left( \frac{\Delta t}{(2\varepsilon + \Delta t\sigma_e)} \right) J^n
\end{align*} \]

And we have the definiton of the curl operator (i.e x axis in this case)

\[ \left( \nabla \times E \right)_x = \frac{dE_z}{dy} - \frac{dE_y}{dz} = \frac{E_{z,i,j,k}^{i,j,k} - E_{z,i,j,k}^{i,j,k-1}}{\Delta y} - \frac{E_{y,i,j,k}^{i,j,k} - E_{y,i,j,k}^{i,j,k-1}}{\Delta z} \]

Now we can make a substitution in a 3D case passing through equations 3 and 4 towards equation 2.
Direct Mapping I

With the purpose of simplify the model we can adjust equation 2, making a general notation, with the following expression:

\[ x_0(n) = \sum_{k=0}^{M} A_k x_k(n - 1) + \sum_{j=0}^{N} \sum_{k=0}^{M} B_j^k x_0(n - j, k) - I_{bias} \] (6)

With this model structure we can use \( I_{bias} = 0 \) because the Maxwell PDE doesn’t have any constant outside.

\[ H_k(n) = \underbrace{\sum_{k=0}^{M} A_k^H H_k(n - 1)}_{\text{H-Field Neighbors (Space)}} + \underbrace{\sum_{k=0}^{M} A_k^M M_k(n - 1)}_{\text{M-Sources Neighbors}} + \underbrace{\sum_{j=0}^{N} \sum_{k=0}^{M} B_j^H E_k(n - j, k)}_{\text{Previous values (Space and Time)}} \] (7)

\[ E_k(n) = \underbrace{\sum_{k=0}^{M} A_k^E E_k(n - 1)}_{\text{E-Field Neighbors (Space)}} + \underbrace{\sum_{k=0}^{M} A_k^J J_k(n - 1)}_{\text{J-Sources Neighbors}} + \underbrace{\sum_{j=0}^{N} \sum_{k=0}^{M} B_j^E H_k(n - j, k)}_{\text{Previous values (Space and Time)}} \] (8)
Direct Mapping II

With equations 7 and 8, we can substitute directly the matrices with following parameters, (We should take into account that we don't have any time displacement in Maxwell PDE, in the case of neighbors).

\[
A_{1}^{E} = \begin{bmatrix}
0 & 0 & 0 \\
0 & \frac{2\epsilon-\Delta t\sigma_{e}}{2\epsilon+\Delta t\sigma_{e}} & 0 \\
0 & 0 & 0
\end{bmatrix} \quad A_{1}^{H} = \begin{bmatrix}
0 & 0 & 0 \\
0 & \frac{2\mu-\Delta t\sigma_{m}}{2\mu+\Delta t\sigma_{m}} & 0 \\
0 & 0 & 0
\end{bmatrix}
\]

\[
A_{1}^{J} = \begin{bmatrix}
0 & 0 & 0 \\
0 & -\frac{\Delta t}{2\epsilon+\Delta t\sigma_{e}} & 0 \\
0 & 0 & 0
\end{bmatrix} \quad A_{1}^{M} = \begin{bmatrix}
0 & 0 & 0 \\
0 & -\frac{\Delta t}{2\mu+\Delta t\sigma_{m}} & 0 \\
0 & 0 & 0
\end{bmatrix}
\]

With the curl operator in equation 5 we could find the following (i.e E Field x axis derivated in terms of z axis)

\[
B_{1}^{E_{x,z}} = \begin{bmatrix}
0 & \frac{\Delta t}{\Delta z(2\epsilon+\Delta t\sigma_{e})} & 0 \\
\frac{\Delta t}{\Delta z(2\epsilon+\Delta t\sigma_{e})} & 0 & \frac{\Delta t}{\Delta z(2\epsilon+\Delta t)} \\
\frac{\Delta t}{\Delta z(\Delta t+\sigma_{e})} & \frac{\Delta t}{\Delta z(\Delta t+\sigma_{e})} & 0
\end{bmatrix}
\]
Following the same method, we can obtain the other matrices involved at each pair of axis. Two matrices are necessary for every pair of axis and fields in the 3D FDTD analysis.

\[
B^E_{1}^{x,y} = \begin{bmatrix}
0 & -\frac{\Delta t}{\Delta y(2\epsilon+\Delta t\sigma_e)} & -\frac{\Delta t}{\Delta y(2\epsilon+\Delta t\sigma_e)} & 0 \\
-\frac{\Delta t}{\Delta y(2\epsilon+\Delta t\sigma_e)} & 0 & -\frac{\Delta t}{\Delta y(2\epsilon+\Delta t\sigma_e)} & 0 \\
-\frac{\Delta t}{\Delta y(2\epsilon+\Delta t\sigma_e)} & -\frac{\Delta t}{\Delta y(2\epsilon+\Delta t\sigma_e)} & 0 & 0 \\
-\frac{\Delta t}{\Delta y(2\epsilon+\Delta t\sigma_e)} & -\frac{\Delta t}{\Delta y(2\epsilon+\Delta t\sigma_e)} & -\frac{\Delta t}{\Delta y(2\epsilon+\Delta t\sigma_e)} & 0
\end{bmatrix} \quad (12)
\]

\[
B^H_{1}^{x,z} = \begin{bmatrix}
0 & \frac{\Delta t}{\Delta z(2\mu+\Delta t\sigma_m)} & \frac{\Delta t}{\Delta z(2\mu+\Delta t\sigma_m)} & 0 \\
\frac{\Delta t}{\Delta z(2\mu+\Delta t\sigma_m)} & 0 & \frac{\Delta t}{\Delta z(2\mu+\Delta t\sigma_m)} & 0 \\
\frac{\Delta t}{\Delta z(2\mu+\Delta t\sigma_m)} & \frac{\Delta t}{\Delta z(2\mu+\Delta t\sigma_m)} & 0 & 0 \\
\frac{\Delta t}{\Delta z(2\mu+\Delta t\sigma_m)} & \frac{\Delta t}{\Delta z(2\mu+\Delta t\sigma_m)} & \frac{\Delta t}{\Delta z(2\mu+\Delta t\sigma_m)} & 0
\end{bmatrix} \quad (13)
\]

Now with approach that has been evaluated in 2D for this notation we can generalize it in 3D as 3D matrices. As you can see diagonal values are always zero then we can reduce the number of wires.
**FDTD+CNN wires connection reduction**

**Figure:** CNN structure that should be used for FDTD, due to the regular structure of the Yee Cells representation [2].
HyperCube Structure

Now as we solved the FDTD for 3D dimension, we have to make a map of this 2D structure on previous figure.

Figure: 3D CNN structure for a specific (i.e. rectangular cavity), there is 64 cells distributed along the cavity [4]
Then we use MPI to map all the data that we have in a hypothetical sequential cavity with incomplete binomial tree scheme.

**Figure:** This is an incomplete binary tree with connections between the base increasing numbers (i.e. $0 \rightarrow 1, 2, 4$ and so on.)
Initialization Stage II

Following the incomplete binary tree we can access to the quantity of packets that could be send from one Cell to another, to share all the data inside the CNN structure.

\[
D_p(N_p) = \sum_{i=0}^{N_{pc}/2} \log_2 \left( \frac{N_{pc}}{2i} \right) - \phi(N_p)
\]  

(14)

\(N_p\) is the number of processor or cells inside the array (for a incomplete binary tree) and \(N_{pc}\) for nearest complete binary tree, and \(\lambda\) and \(\beta\) represent the delay and the bandwidth, involved per each transmission/reception. After the initialization stage we have a computation process that parallelize the 3D structure per pairs along the cavity.
**φ Function**

The $\phi$ function makes the complete binary tree number of packets model coherent, given by the expression $\sum_{i=0}^{N_{pc}/2} \log_2 \left( \frac{N_{pc}}{2^i} \right)$, this variation is a linear increasing behavior until the nearest 2-power bases.

**Figure:** This is the $\phi$ function that make ideal the incomplete binary tree mode $T_x/R_x$ patterns.
Sequential/Parallel Time Approach I

The sequential execution time was calculated using the sequential program in C.

Figure: The spatial decomposition shows a third degree polynomial increasing in terms of time and series of polynomial fitting functions in terms of number of timesteps variation.
The timesteps affect the FDTD time execution in a linear way, and varies its mathematical slope.

**Figure:** The simulation timesteps density describes a linear increasing behavior on computational complexity.
Sequential/Parallel Time Approach III

First, we have that only one spatial three nested cycles. Then, we obtain a single measure $t_{seq} = 36.56 \times 10^{-3} \ [s]$, after that we substitute expression 14 as number of packets sent in this context and we could obtain a theoretical speedup threshold for our cavity app.

$$\frac{T_{cs}}{t_{seq}} = \left[ \frac{\tau}{\Delta t} \right] t_{seq}$$ (15)

$$T_p = \left[ \sum_{k=0}^{pq} D_p(k, N_p) \left( \lambda + \frac{D_n}{\beta} \right) \frac{N_p \zeta}{N_p \alpha} \right] + \left[ \frac{\tau}{\Delta t} \right] + \sum_{k=0}^{pq} D_p(k, N_p) \left( \lambda + \frac{D_n}{\beta} \right)$$ (16)

Then with the parallel structure we can estimate the parallel time $T_p$, with the number of packages sent, the backplane limitations ($\lambda$ and $\beta$) and a $\zeta, \alpha$ variables that are a resulting random normal distributions fixed for MPI scheduler execution ($0 \leq \zeta \leq 1, 0 \leq \alpha \leq 1$), $D_n$ variable is the number of bytes inside each packet.
Figure: This pattern communication for binomial tree and the pair computational communication scheme increase in terms of number of cells, and should decrease by half per each cell initialized.
Figure: Speedup has a dominant communication part at the lower quantity of cells, and an initialization time that is critical when the number of cells increases (This values are taken from the specific cavity studied here).

\[ Sp(T) = \frac{\sum_{k=0}^{pq} D_p(k, N_p) \left( \lambda + \frac{D_n}{\beta} \right)}{N_p \zeta t_{seq}} + \frac{1}{N_p \alpha} + \frac{\Delta t}{\tau t_{seq}} \sum_{k=0}^{pq} D_p(k, N_p) \left( \lambda + \frac{D_n}{\beta} \right) \]
Cavity Description I

The following figure express the rectangular cavity that was used in WCT simulator EM-software.

**Figure:** Rectangular Cavity representation in WCT EM-simulator.
Cavity Description II

To describe the rectangular cavity we have the following parameters

**Physical IS units:**
- length $\rightarrow$ millimeters, time $\rightarrow$ seconds, frequency (sources) $\rightarrow$ Hertz

**Cavity Parameters:**
- Cavity lengths $\rightarrow$ from $[0.5 \ 0.5 \ 0.5]$ to $[20.5 \ 16.5 \ 7.5]$  
- Relative Permittivity $\epsilon_r = 2$ and Relative Permeability $\mu_r = 1$  
  Homogenous material with PEC in the frontiers.
- **Source:** Electric dipole with polarization vector $[0 \ 0 \ 1]$, BHW (Blackman Harris Window) order 0, $F_c = 7 \times 10^9$ $[Hz]$, position $[2.5 \ 7.5 \ 4.5]$.
- **Observer:** position $[7 \ 3 \ 4]$, $\Delta t = 1 \times 10^{-12}$ $[s]$, and total simulation time $T = 2.001 \times 10^{-9}$ $[s]$.
- **Spatial Decomposition:** $\Delta x = \Delta y = \Delta z = 5 \times 10^{-3}$ $[mm]$, the number of Yee Cells in the array is $41 \times 33 \times 15 = 20295$. 


$E_y$ Field time results, then you can see the DC component that is involved in the BHW order 0 spectrum.

**Figure:** Time Results for the observer position for a total time step $2 \times 10^{-9}$ [s].
$H_x$ results, we can see a component the belongs to the source characteristic frequency $f_c = 7 \times 10^9$ [Hz], and some resonances with higher frequency according to the resonant cavity model.

**Figure:** Time Results for the observer position for a total time step $2 \times 10^{-9}$ [s].
Time domain error for E fields.

**Figure:** Absolute Error Error for E Fields in the Cavity.
Time Domain Absolute Error II

Time domain error for H fields.

**Figure:** Absolute Error Error for H Fields in the Cavity.
Taking into account the previous transient results we have the following average results, for the considerable and stable $E$ and $H$ fields on the cavity.

**Table:** Average Absolute Error Results Field v.s Simulators

<table>
<thead>
<tr>
<th>Fields / Simulators</th>
<th>Meep</th>
<th>OpenEMS</th>
<th>WCT</th>
<th>Octave (Custom App)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ex</td>
<td>0.7903%</td>
<td>0.8233%</td>
<td>0.1875%</td>
<td>0.6590%</td>
</tr>
<tr>
<td>Ey</td>
<td>1.117%</td>
<td>1.1824%</td>
<td>0.3512%</td>
<td>0.8224%</td>
</tr>
<tr>
<td>Ez</td>
<td>4.526%</td>
<td>3.5575%</td>
<td>1.4928%</td>
<td>2.4548%</td>
</tr>
<tr>
<td>Hx</td>
<td>2.5892%</td>
<td>1.5766%</td>
<td>2.5247%</td>
<td>2.8954%</td>
</tr>
<tr>
<td>Hy</td>
<td>5.6264%</td>
<td>5.5303%</td>
<td>3.1005%</td>
<td>2.4707%</td>
</tr>
</tbody>
</table>
Conclusions

- CNN is a computational structure that gives the possibility to parallelize a regular numerical domain, such as FDTD spatial domain and accelerate its time execution. For our case we have some maximum threshold around 60-200 processors that give us the possibility to work with not so long quantity of processors and obtain more 20, 30 (or more if we improve the backplane times) times total execution improvement.

- 3D computational scheme is a disadvantage in terms of communication scenario, but inside the speedup range the transient results are considerably similar with the sequential application, taking into account communication overload by non-deterministic pairs (MPI).

- CNN is flexible for any kind or device or computational paradigm due to its connection and programming flexibility and EM problems aren’t an exception.
Future Work

- Implement this scheme on FPGA and find a experimental speedup thershold that belong to theoretical range is short term task that we have.

- Develop an application with lumped elements or oblique-irregular structures with different physical properties is another challenged in mid-long term that we expect that could give more parameters to join CNN with FDTD method.
References


Thanks for your attention!!!

E-mails:

- jmmayor@javerianacali.edu.co
- let12@duke.edu
- nagyz@szaki.hu
- tek@javerianacali.edu.co