

# A MLP Based FDTD Method

Shuangyang Li, Shengjian Lai, Jinhao Liu, Shaobo Wu, Lang Chen

CEM Lab, School of Physics, University of Electronic Science and Technology of China, Chengdu, China

Email: syli@std.uestc.edu.cn

**How to cite this paper:** Li, S.Y., Lai, S.J., Liu, J.H., Wu, S.B. and Chen, L. (2020) A MLP Based FDTD Method. *Journal of Computer and Communications*, 8, 279-284.  
<https://doi.org/10.4236/jcc.2020.812022>

**Received:** November 20, 2020

**Accepted:** December 24, 2020

**Published:** December 31, 2020

## Abstract

In this paper, we proposed a novel multilayer perceptron (MLP) based Finite-Difference Time-Domain (FDTD) method to reduce the time complexity of the conventional FDTD method, MLP neural networks can be used to replace the field quantities update equations and we found that in certain scenario, we can greatly reduce the time complexity of FDTD method.

## Keywords

FDTD, Neural Network, MLP

## 1. Introduction

Machine learning has become a cornerstone of modern science since it can be applied in all areas of science [1]. However, its use in computational electromagnetics (CEM) is not yet wide.

In this paper, we propose a new FDTD algorithm that combines machine learning techniques with conventional FDTD algorithms in CEM; and, we will see that it can effectively reduce the time complexity of conventional FDTD algorithms in certain scenarios.

## 2. FDTD Method

The conventional FDTD Method in CEM can be described as follow. First, we use Maxwell's equations:

$$\begin{cases} \frac{\partial \mathcal{D}}{\partial t} = \nabla \times \mathcal{H} \\ \frac{\partial \mathcal{B}}{\partial t} = -\nabla \times \mathcal{E} \end{cases} \quad (1)$$

Assume a 2D uniform square grid case with TM mode, which means  $E_x = E_y = H_z = 0$ , applying central difference scheme to it, Equation (1) can be written as:

$$\left\{ \begin{aligned} H_x^{n+\frac{1}{2}}(i, j + \frac{1}{2}) &= H_x^{n-\frac{1}{2}}(i, j + \frac{1}{2}) - \frac{E_z^n(i, j + 1) - E_z^n(i, j)}{\Delta y} \\ H_y^{n+\frac{1}{2}}(i + \frac{1}{2}, j) &= H_y^{n-\frac{1}{2}}(i + \frac{1}{2}, j) + \frac{E_z^n(i + 1, j) - E_z^n(i, j)}{\Delta x} \\ E_z^{n+1}(i, j) &= E_z^n(i, j) + \left[ \frac{H_y^{n+\frac{1}{2}}(i + \frac{1}{2}, j) - H_y^{n+\frac{1}{2}}(i - \frac{1}{2}, j)}{\Delta x} \right. \\ &\quad \left. - \frac{H_x^{n+\frac{1}{2}}(i, j + \frac{1}{2}) - H_x^{n+\frac{1}{2}}(i, j - \frac{1}{2})}{\Delta y} \right] \end{aligned} \right. \quad (2)$$

where in Equation (2)  $n$  represents time steps and  $i, j$  represents spatial coordinates, and thus we can calculate the electromagnetic field quantities of next time step by using the previous field quantities. Solving electromagnetic problems by using conventional FDTD method can be described as follows (Figure 1).

As mentioned above, with determined absorbing boundary conditions (ABC) and excitation sources, we can calculate the electromagnetic field quantities using this “leapfrog integration” technique, which uses electric fields to calculate magnetic fields, and magnetic fields to calculate electric fields. While neural networks can learn a non-linear function that maps an input to an output based on example input-output pairs [2], this allows us to feed electromagnetic field quantities of previous time step as input and predict the electromagnetic field quantities of next time as we do in the conventional FDTD method. Here are the general steps to combine neural network with conventional FDTD method: 1) Collect Data From conventional FDTD Method; 2) Train MLP network using our collected data; 3) Combine MLP network with FDTD method.

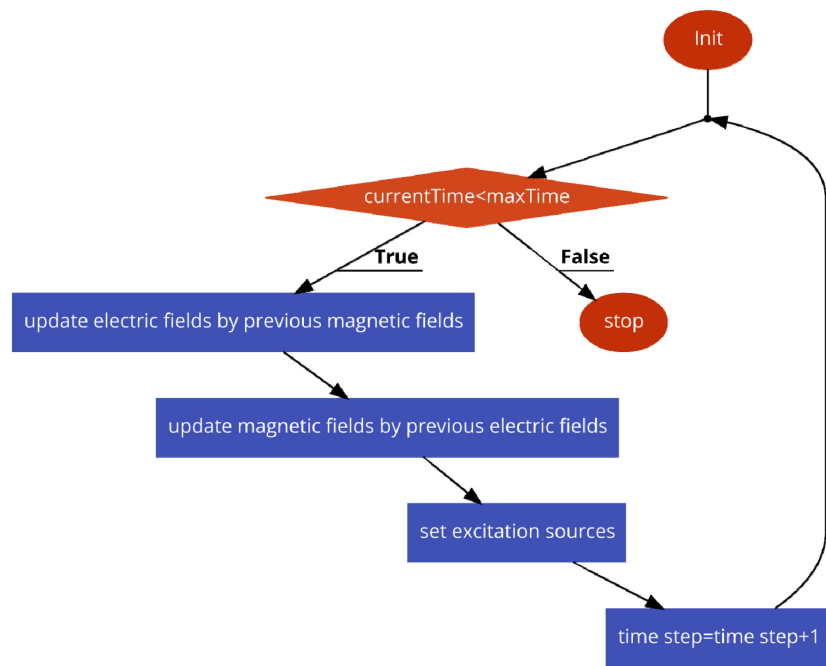


Figure 1. Conventional FDTD method.

### 3. MLP Model

MLP neural network is one of the most widely used neural networks in modern machine learning techniques. As a universal approximator [3], MLP is suitable for our goal. Its architecture in our case is shown in **Figure 2**. Its governing equation in our case can be written as:

$$O_r = G(wI_r + b) \quad (3)$$

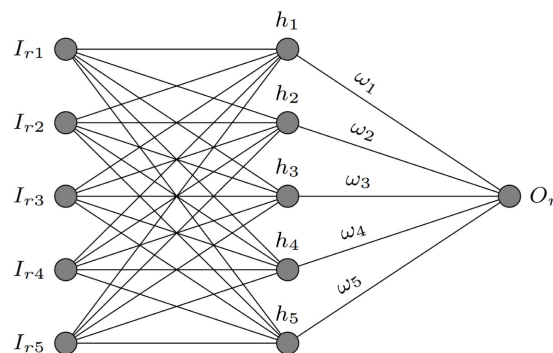
where  $w$  is the weight of the neural network and  $b$  is the bias of the neural network, function  $G$  represents the neural network architecture, here we take the input vector as  $I_r = [E_z^n(i, j), H_y^{n+1/2}(i+1/2, j), H_y^{n+1/2}(i-1/2, j)]$ , and the output vector as  $O_r = [E_z^{n+1}(i, j)]$ . Then we can calculate the electric field quantities of next time step using the MLP neural network predicted value, and calculate the magnetic field quantities of next time step using the MLP neural network predicted value, thus we can replace the conventional FDTD update equations with our trained neural networks.

Another thing we have found is that with certain boundary conditions, we can jump out of the conventional “leapfrog integration” techniques, which use electric fields to calculate magnetic fields, and magnetic fields to calculate electric fields. We can calculate the electric fields of next time step using only electric fields of previous time, and no magnetic fields are involved. If we change the input to  $I_r = [E_z^n(i+1, j), E_z^n(i-1, j), E_z^n(i, j+1), E_z^n(i, j-1), E_z^n(i, j)]$ , which is the five electric field points in mesh grid around the output point. When using 2D Mur first-order absorbing boundary condition in TM mode, it can be written as:

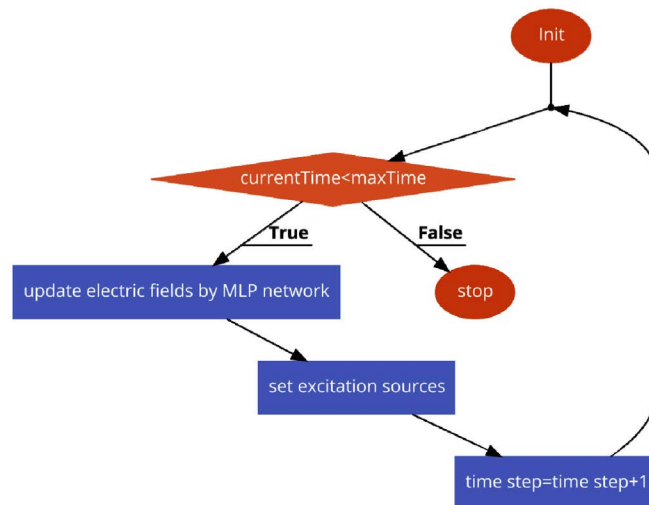
$$\left[ \frac{\partial E_z}{\partial x} - \frac{1}{c} \frac{\partial E_z}{\partial t} \right] \Big|_{x=0} = 0 \quad (4)$$

where no magnetic fields are involved, we can apply this boundary condition to our FDTD method, and it can be simplified as follow **Figure 3**.

Because we skipped the magnetic fields quantities update equations in the improved machine learning-based FDTD method mentioned above, we can say that we reduced the algorithmic complexity of FDTD method by using the proposed machine learning-based FDTD method in certain scenario, but we have to



**Figure 2.** MLP neural network architecture in our case.



**Figure 3.** Simplified FDTD method.

mention that by using neural networks we introduced extra artifacts that may lead to less accurate numerical results. In the MLP neural network architecture, we can achieve an average relative error around 13% at best.

#### 4. Simulation Results and Discussion

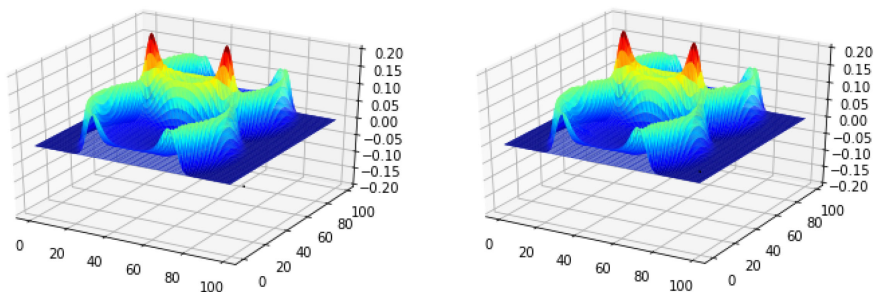
In order to examine the accuracy of our newly proposed method, we have made some numerical simulations using python programming language along with keras machine learning framework. We used a 2D 100 \* 100 square mesh grid as our simulation area. We set the temporal offset  $\Delta t$  to 0.1 and spatial offset  $\Delta x = \Delta y = \sqrt{2}c\Delta t$  to satisfy the Courant-Friedrichs-Lewy (CFL) condition. The boundary conditions are set to 2D Mur first-order absorbing boundary condition and two Gaussian pulses are placed at point (25, 50) and (75, 50) as our excitation sources. The max time step is set to 100.

Comparisons of conventional FDTD method generated  $E_z$  surface plot vs neural networks generated  $E_z$  surface plot are shown in **Figure 4**. We also take a number of sample points to observe the difference in waveforms of individual points., which are shown in **Figure 5** and **Figure 6**.

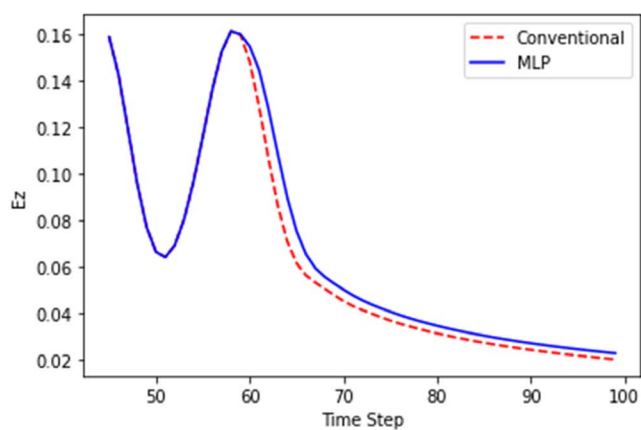
As the results indicate that the MLP network can solve electromagnetic problems with acceptable accuracy, the MLP model can achieve an average relative error around 13%, while in certain scenario (e.g. 2D Mur first-order ABC), We can completely omit the equation for updating the magnetic field by using MLP neural network, which will greatly reduce the time complexity of FDTD method.

#### 5. Conclusion

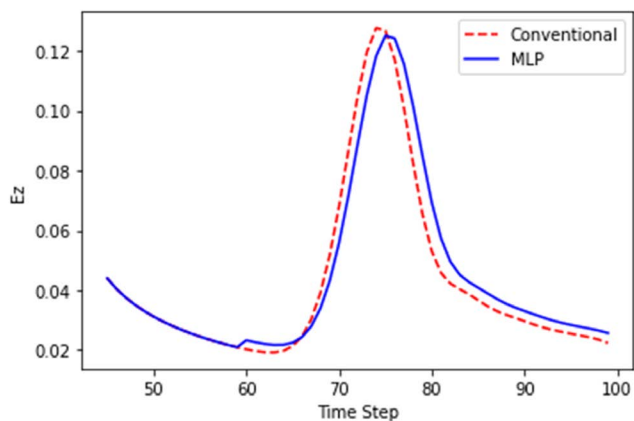
In this paper, we propose a novel FDTD algorithm based on MLP neural networks and find that it can effectively reduce the time complexity of conventional methods under certain conditions. The integration of machine learning into conventional CEM still has a long way to go, but we believe that it's the future for CEM.



**Figure 4.** Simulation results comparison between conventional FDTD method (left) and MLP (right).



**Figure 5.** Waveform comparison between conventional FDTD method (Red) and MLP (Blue) at point (70, 70).



**Figure 6.** Waveform comparison between conventional FDTD Method (Red) and MLP (Blue) at point (45, 45).

### Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

### References

- [1] Krizhevsky, A., Sutskever, I. and Hinton, G. (2012) ImageNet Classification with

Deep Convolutional Neural Networks. NIPS 2012: Neural Information Processing Systems.

- [2] Stuart, J. and Russell, P.N. (2010) Artificial Intelligence: A Modern Approach. 3rd Edition, Prentice Hall.
- [3] Pinkus, A. (1999) Approximation Theory of the MLP Model in Neural Networks. *Acta Numerica*, **8**, 143-195. <https://doi.org/10.1017/S0962492900002919>