Learning and Compensating Numerical Dispersion Errors in FDTD with Artificial Neural Networks

Xingqi Zhang, Aristeidis Seretis, and Costas D. Sarris
The Edward S. Rogers Sr. Dept. of Electrical & Comp. Engineering
University of Toronto, Toronto, ON, Canada

The remarkable advances in the available computational power over the past few years, and those anticipated to come, have propelled machine learning algorithms, some developed decades ago, to the forefront of research interest in a wide and diverse range of fields: from medicine to autonomous vehicles and robotics. As the interest in this area deepens, new algorithmic and theoretical developments are reported and additional applications are explored. This trend has been extended to computational science and engineering, with machine-learning algorithms designed to either improve (D. Ray and J. Hesthaven, *J. Comput. Phys.* 367 (2018), pp 166-191) or even replace (K. Mills et al., *Phys. Rev. A*, vol. 96, 2017) existing numerical solvers for linear and nonlinear problems.

In this work, we follow the former route to explore machine learning algorithms for numerical dispersion compensation in the Finite-Difference Time-Domain (FDTD) method. A modular deep neural network (MDNN) is trained with FDTD simulations of varying cell size, with the goal of "learning" the pattern of numerical dispersion errors by comparing solutions of various two and three-dimensional problems at coarse and dense grids. Hence, our training data include not only a wide collection of geometries, but also meshes of variable density for each problem. We present a thorough analysis of the structure of this MDNN and its error performance as a function of training data. We evaluate its ability to act as a numerical dispersion compensation engine: one that when fed with the results of a coarse mesh FDTD simulation can predict the results of an FDTD simulation in a mesh uniformly refined by an integer factor N.

In terms of training, two approaches are presented and compared: first, a general purpose one, whereby the network is trained by a large collection of problems and is asked to produce the FDTD results in a refined mesh for a problem it has not seen before; second, on-the-fly training where the coarse and dense simulations are concurrently run for a number of time steps, until the ANN is trained to use only the coarse FDTD data to predict the dense FDTD data, for this specific geometry. The comparative presentation of these approaches enables the illustration of their relative advantages and disadvantages.