DeepSim-HiPAC: Deep Learning High Performance Approximate Calculation for Interactive Design and Prototyping

Ahmed Al-Jarro
Fujitsu Laboratories of Europe Ltd., Hayes Park Central, Hayes End Road, Hayes UB4 8FE, Greater London, UK
ahmed.al-jarro@uk.fujitsu.com

Serban Georgescu
Fujitsu Laboratories of Europe Ltd., Hayes Park Central, Hayes End Road, Hayes UB4 8FE, Greater London, UK
serban.georgescu@uk.fujitsu.com

Yasumoto Tomita
Fujitsu Laboratories Ltd., 4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki, Kanagawa 211-8588, Japan
tomita.yasumoto@jp.fujitsu.com

Kouta Nakashima
Fujitsu Laboratories Ltd., 4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki, Kanagawa 211-8588, Japan
nakashima.kouta@jp.fujitsu.com

ABSTRACT

We present a data-driven technique that can learn from physical-based simulations for the instant prediction of field distribution of three-dimension (3D) objects. Such techniques are extremely useful when considering, for example, computer aided engineering (CAE), where computationally expensive simulations are typically required. To accelerate this process, we propose a deep learning framework that can predict the principal field distribution given a 3D object. This work allows us to learn a systems' response using simulation data of arbitrarily shaped objects and an auto-encoder inspired deep neural network that maps the input of the 3D object shape to its principal 3D field distribution. Here, we demonstrate the effectiveness of our technique to aid the prediction of simulation results for two distinctive applications: micro-magnetics design in computational electromagnetics (CEM) and interactive cooling systems design in computational fluid dynamics (CFD). We show that our convolutional neural network engine, called DeepSim-HiPAC, can approximate field distribution several orders of magnitude faster, up to 250000X, and at a cost of low error rate.

CCS CONCEPTS

• Computer methodology → Artificial intelligence; Deep neural networks • Applied computing → Computer-aided engineering

KEYWORDS

Artificial intelligence, Machine learning, Deep neural networks, High performance approximate calculation, HPC, CEM, CFD

ACM Reference format:

Ahmed Al-Jarro, Serban Georgescu, Yasumoto Tomita and Nakashima Kouta. 2018. DeepSim-HiPAC: Deep Learning High Performance Approximate Calculations for Interactive Design and Prototyping. *In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2018, Dallas, TX, USA, 2 pages.* https://doi.org/10.1145/XXXXXXXXXX

SC'18, November, 2018, Dallas, Texas USA

https://doi.org/10.1145/ XX.XXXX/XXXXXXXXXX

1 Introduction

After decades of slow progress in artificial intelligence (AI), the recent advancements in High Performance Computing (HPC) and the development of sophisticated deep neural network (DNN) architectures have given AI the critical mass it needed for fueling the so-called next wave of technological revolution. The training process of DNNs can be an extremely computationally intensive task, which typically requires large computational resources. Several parallelization schemes [1-4] have been proposed for the speedup of DNNs while exploiting the computational power of both specialized and mainstream hardware accelerators; including, for example, GPUs, Xeon Phi, FPGA and ASICs. These recent advancements in AI, and its speedup, have readily demonstrated its potential to deliver endless possibilities in a wide range of scientific and economic developments; some of them are on the verge of disrupting the evolution of the socio-technological eco-system we know today. Notably, the recent development of DNNs has demonstrated significant commercial success to speech, object recognition and classification with accuracy performance exceeding those recorded by other methods in machine learning.

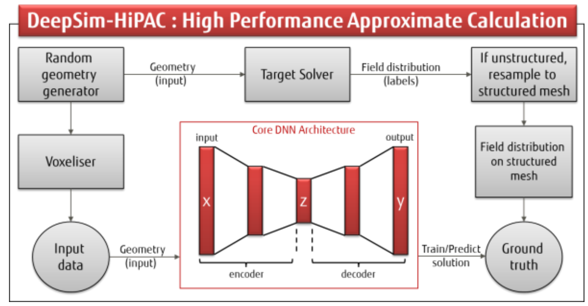
On the back of the recent advancement in AI, much recent effort is observed in experimenting with neural networks to predict the outcome of physical-based processes [5, 6]. Numerical simulations are regarded the most effective way to understand complex physical phenomena. Traditional numerical simulation usually requires extensive computing resources; whereas, neural networks may provide a less accurate, but much faster and more efficient solution. In this work, we propose a deep learning framework, here referred to as DeepSim-HiPAC, that can predict the principal field distribution given a three-dimension (3D) object. We achieve this by feeding steady state simulation data into deep neural networks so they can learn and predict field distribution of arbitrarily shaped 3D objects. We validate our method for two different use cases of distinctive physics. The results show that the neural network can predict the distribution of fields as an output given only a definition to the target 3D geometry as an input. It also implicitly learns the subtle differences between different geometrical shapes.

2 Methodology & Results

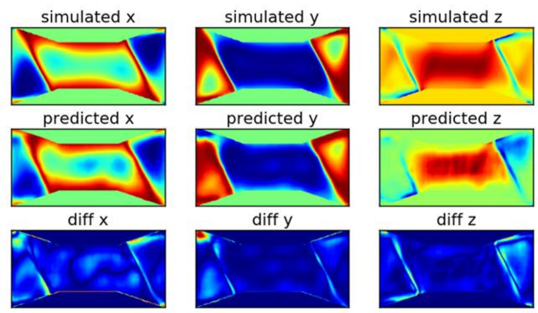
The structure to our DeepSim-HiPAC engine and some exemplary results are shown in Figure 1. In a) we outline the workflow to data generation as well as network training and results prediction. In our engine, we use an auto-encoder inspired convolutional neural network [7] to map the input of the 3D object shape to the output data frame that represents the principal three-dimensional field distribution. We train the neural networks in DeepSim-HiPAC by feeding it thousands of records, each representing an arbitrarily shaped 3D input geometry and corresponding 3D ground truth simulations, all obtained for the same physical conditions. No physical equations or numerical approximation methods are fed into the neural network. The training data and test data are generated for two distinctive applications: i) micro-magnetics design in computational electromagnetics (CEM) and ii) interactive cooling systems design in computational fluid dynamics (CFD). For the data generation, we use simulation codes that solve the Landau-Lifshitz-Gilbert and Navier-Stokes equations for cases i) and ii), respectively. The generation of training data is obtained by feeding the set of arbitrarily shaped 3D geometries into the target solver in order to produce corresponding 3D steady state field distribution. This time-consuming data generation stage is a preprocessing step, which we perform on a HPC cluster prior to training. We evaluate our technique for the two different use cases. In micro-magnetics, case i), we consider the same conditions as those of the so-called standard problem 4, SP4, [8] whereby a rectangular sample with the material parameters of permalloy is prepared in an S-state. Then the time evolution of the system under the influence of an external field is computed until steady state that we aim to learn. In CFD, case ii), we consider similar conditions to those found in a server cooling scenario, whereby air enters the domain from an inlet at the left, passes between (and/or around) the fins of a heatsink and leaves the domain through a smaller outlet on the right. The iterative evolution of the system is computed until convergence and the steady state is recorded that we aim to learn.

For the training stage, we use the auto-encoder inspired neural network and a structured grid end-to-end mapping between the input (x) and output (y) that are of the same size. We train the convolutional neural network using the collection of randomly shaped 3D geometries, input (x), and corresponding steady state field distribution, output (y), and evaluate the prediction over a validation dataset of similar profile to the training dataset and a test dataset of 3D shapes that are of different profile. In Figure b) we illustrate the quality of prediction for a cross-sectional plane at the middle of the z-axis of a 3D extruded bow-shaped geometry for case i). The top panel shows the ground truth simulations (*S*), whereas the middle panel shows those predicted (*P*) by DeepSim-HiPAC engine. The bottom panel plots the absolute difference between the two sets of results across all principal field components, i.e. diff = |*P* - *S*| for x, y and z fields. The average of the collective relative error, across the combined three principal fields, is $1/3n\sum_i^{3n}\left|P^i_{(x,y,z)} - S^i_{(x,y,z)}\right| = 0.0338$, where n is the size of y. In c) we illustrate the agility of DeepSim-HiPAC when applying it in the interactive design for case ii). Here we interactively adjust the position of the components within the domain as desired, where the 3D flow prediction is updated instantly on demand (a → b → c → d).

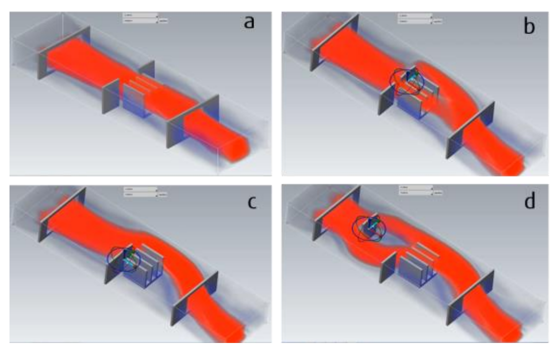
We demonstrate our technique to provide immediate feedback for real-time and interactive design iterations at the initial stage where extremely high fidelity results are not mandatory; enabling extremely rapid predictions for dynamic design and prototyping. As showcased in this work, the attraction of this technique is its applicability to learn a wide range of tasks benefiting many fields in engineering design and prototyping problems with extremely fast response. Our Initial speedup performances for DeepSim-HiPAC are recorded up to 250000 times faster than those directly calculated by the native (target) solvers and at a low error rate.



a) DeepSim-HiPAC: workflow to data generation, neural network training and prediction



b) Prediction of micro-magnetics fields in CEM



c) Interactive design for cooling system in CFD

Figure 1: Workflow to data generation, neural network training and results prediction

REFERENCES

[1] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Y. Ng (2012). Large Scale Distributed Deep Networks. In Advances in Neural Information Processing Systems, 1223-1231.

[2] A. Coates, B. Huval, T. Wang, D. J. Wu, A. Y. Ng, and B. Catanzaro (2013). Deep Learning with COTS HPC Systems. In Proceedings of the 30th International Conference on Machine Learning, 1337–1345.

[3] A. Krizhevsky (2014). One Weird Trick for Parallelizing Convolutional Neural Networks. arXiv preprint arXiv:1404.5997.

[4] S. Aldea, A. Al-Jarro, B. Bhaskaran, T. Tabaru, and A. Ike (2017). Novel Distribution of Memory Intensive Fully Connected Layers in Multi-GPU Deep Learning. In Proceedings of NVIDIA GPU Technology Conference. Top 20 poster.

[5] J. Tompson, K. Schlachter, P. Sprechmann, and K. Perlin (2016). Accelerating Eulerian Fluid Simulation With Convolutional Networks. arXiv preprint arXiv:1607.03597.

[6] X. Guo, W. Li, and F. Iorio (2016). Convolutional Neural Networks for Steady Flow Approximation. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 481–490.

[7] J. Schmidhuber (2015). Deep Learning in Neural Networks: An Overview. Neural Networks 61, 85–117. arXiv preprint arXiv:1404.7828.

[8] Micromagnetic Modeling Activity Group - µMag, National Institute of Standards and Technology, Gaithersburg, MD (2008), https://www.ctcms.nist.gov/~rdm/std4/spec4.html.