The Journal of Open Source Software

Staidy: a python package for AI-based prediction of steady-state fields for finite difference applications

Octavi Obiols-Sales[1], **Aparna Chandramowlishwaran**[1], **Abhinav Vishnu**[2], **and Nicholas Malaya**[2]

1 University of California, Irvine **2** Advanced Micro Devices (AMD)

DOI: DOIunavailable

Software

- Review
- Repository
- Archive

Editor: Pending Editor

Reviewers:

- @Pending Reviewers

Submitted: N/A
Published: N/A

In partnership with

AMERICAN ASTRONOMICAL SOCIETY

This article and software are linked with research article DOI 10.3847/xxxxx, published in the Astrophysical Journal <- The name of the AAS journal..

Summary

Deep learning (DL) algorithms have demonstrated remarkable improvements in a variety of modeling/classification tasks such as computer vision, natural language processing, and high-performance computing (Baldi et al., 2014; Krizhevsky et al., 2012; Liu et al., 2016). At the same time, several researchers have recently applied DL methods for modeling physical simulations that are usually resolved by finite difference methods (Guo et al., 2016; Obiols-Sales et al., 2020; Raissi et al., 2019).

Obiols-Sales et al. (2020) presented CFDNet, a deep-learning based accelerator for steady-state fluid simulations. CFDNet accelerates traditional CFD solvers by placing a convolutional neural network (CNN) at the core of steady-state simulations: the CNN takes as input an intermediate field and outputs the corresponding steady-state field. Then, this CNN-predicted steady-state field is fed back into the physics solver, which constraints the solution in few iterations. The framework showed promising results with coarse-grid (low-resolution) simulations, but is computational impractical for fine-grid (high-resolution) simulations because of the unsormountable data collection and training time. SURFNet is an extension of CFDNet that targets high-resolution simulations easing the mentioned computational burdens. SURFNet enhances CFDNet by transfer learning the weights calibrated with coarse-grid solutions to high-resolution settings. This enables a 15x smaller dataset collection of high-resolution solutions.

Statement of need

`Staidy` is a Python package for dataset generation, CNN training, CNN prediction, and transfer learning between different grid resolutions of steady-state solutions, targetted to any finite difference application usually found in scientific computing. `Staidy` was designed to provide a general recipe for CNN-based acceleration of finite difference solvers by harnessing the data generated from these solvers without any domain/practitioner intervention. That is, `staidy` is amenable from computational fluid dynamics to solid mechanics, passing through heat transfer problems. `Staidy` contains four critical functionalities. First, dataset generation according to the input-output representation in Obiols-Sales et al. (2020). Second, CNN setup and training. Third, CNN-based prediction of steady-state fields and its quantitative evaluation. And four, transfer learning the weights calibrated with coarse-grid solutions for alleviating the CNN training time of fine-grid data.

`Staidy` was designed for applicability and reproducibility of the results of CFDNet (Obiols-Sales et al., 2020) and SURFNet. It can be used by both (a) domain practitioners who wish to accelerate their steady-state applications, and (b) artificial intelligence engineers who aim at network tuning and/or evaluation and enhancement of the network's learning task for physical applications.

Acknowledgements

We acknowledge contributions from NSF/AMD.

References

Baldi, P., Sadowski, P., & Whiteson, D. (2014). Searching for Exotic Particles in High-Energy Physics with Deep Learning. *Nature Commun.*, *5*, 4308. https://doi.org/10.1038/ncomms5308

Baldi, P., Sadowski, P., & Whiteson, D. (2014). Searching for Exotic Particles in High-Energy Physics with Deep Learning. *Nature Commun.*, *5*, 4308. https://doi.org/10.1038/ncomms5308

Guo, X., Li, W., & Iorio, F. (2016). *Convolutional neural networks for steady flow approximation*. 481–490.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 25* (pp. 1097–1105). Curran Associates, Inc. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 25* (pp. 1097–1105). Curran Associates, Inc. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Liu, Y., Racah, E., Prabhat, Correa, J., Khosrowshahi, A., Lavers, D., Kunkel, K., Wehner, M., & Collins, W. (2016). Application of deep convolutional neural networks for detecting extreme weather in climate datasets. *arXiv Preprint arXiv:1605.01156*.

Obiols-Sales, O., Vishnu, A., Malaya, N., & Chandramowlishwaran, A. (2020). CFDNet: A deep learning-based accelerator for fluid simulations. *arXiv Preprint arXiv:2005.04485*.

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, *378*, 686–707.