End-to-End Differentiable Modeling and Management of the Environment

Christopher Krapu[1] and Tyler Felgenhauer[2]

[1] *GeoAI Group, Oak Ridge National Laboratory, Oak Ridge, TN, USA*
[2] *Department of Civil and Environmental Engineering, Duke University, Durham, NC, USA*

Focal Area

(2) Data acquisition and assimilation enabled by machine learning, AI, and advanced methods including experimental/network design/optimization. We emphasize the importance of leveraging optimization techniques from AI/machine learning (ML) to solve challenging problems in Earth system modeling.

Science Challenge

Automatic differentiation has had a transformative effect on ML by allowing the calculation of gradients of arbitrary functions in an incredibly large class of models. We can potentially realize similar improvements in parameter estimation and control for Earth system models (ESMs) by reimplementing them in computational frameworks from ML. Practitioners working with large (>10^7 parameters) models in ML and AI can obtain good predictive performance in a range of spatiotemporal tasks by making use of optimization via stochastic gradient descent and incorporating prior knowledge at multiple levels. We propose writing ESMs in open-source computational frameworks such as Torch, Tensorflow, and JAX to greatly expand the scope of environmental forecasting and management challenges, which can be addressed by leveraging automatic differentiation and gradient descent-like algorithms. We do not call for a wholesale replacement of physical models with data-driven surrogates, but rather advocate for interleaving physical and empirical equations in a manner that is most faithful to the extent of our scientific knowledge and observational data. Central to this topic is the merging of differentiable physical simulations with differentiable optimization layers, which are now both beginning to come to the forefront.

Rationale

This white paper describes a scenario to highlight a few of the essential complexities currently faced by environmental analysts. It is now conceivable that multiple nations could potentially gain the capability to modify the global climate through solar geoengineering. If this happens, the likely technology would use high-altitude aircraft to inject sulfate-based aerosols into the stratosphere, reducing net radiative forcing by reflecting some sunlight back into space. Such a policy, if conducted naively, could potentially lead to disruption of atmospheric hydrologic processes, including moisture transport, cloud formation, and precipitation with possible concomitant impacts on extreme hydrologic events. A wise decision-maker would seek to maximize the beneficial cooling effect of the proposed geoengineering scheme while minimizing ecological and agricultural disruption. Although we have illustrated a particular example, our conclusions hold for highly parameterized physical models of all sorts, including those used to simulate surface and subsurface hydrology.

The aforementioned scenario is simple to formulate at a conceptual level, but it would represent an exceedingly difficult problem from an optimization perspective. We specifically focus on the chief difficulties of scale and uncertainty. Many thousands or even millions of viable spatiotemporal aerosol delivery coordinates exist, each of which may have a unique cost function

DOI: https://doi.org/10.2172/1769703
DOE BER, AI4ESP 2021 #AI4ESP1072 (https://www.ai4esp.org)

associated with its usage. We are further hampered by a less-than-perfect knowledge of the system, as well, with substantial uncertainties associated with model parameters and initial conditions. Thus, we are confronted with many of the challenges associated with stochastic optimization. Given that each climate model forward simulation is quite computationally intensive, the usage of large numbers of forward evaluations to either apply evolutionary algorithms or finite-difference sensitivity methods is not viable. A common alternative is to use data-driven function approximators such as neural networks or Gaussian processes, which have no built-in knowledge of system dynamics and cannot be relied upon to generalize beyond the extent of the data and priors used to estimate their parameters. Below, we explain how using automatic differentiation as implemented in modern ML frameworks can be used to resolve both problems.

Narrative

Large-scale training of neural networks has been made dramatically more effective over the past several decades by the development of techniques for algorithmically computing gradients of functions in computation time, which is constant with regard to the number of inputs. Put simply, autodiff (automatic differentiation) allows us to obtain the entire gradient of a function with only a single function evaluation. This is unlike numerical differentiation, which requires a forward pass for each dimension of the input; it also improves upon symbolic differentiation by avoiding the exponential number of terms that result from naïve applications of the chain rule for derivatives. Additionally, examples of physical models (Hu et al., 2020; Schenck and Fox, 2018) are now implemented in autodiff-enabled programming frameworks. Because the composition of differentiable functions is also differentiable via the chain rule, these forms could be combined and composed to create novel systems amenable to the application of gradient descent or its stochastic counterpart. A novel and intriguing addition to this variety of model components is the differentiable optimization layer in which, for example, the entire procedure of quadratic programming can be treated as a single differentiable function, thus allowing backpropagated gradients to easily flow through (Amos and Kolter, 2017). This would be immensely useful in fashioning integrated socioenvironmental models that allow for interplay between physical dynamics and optimization pursued by managers.

Specifying a highly parameterized model is often much more straightforward than obtaining statistically justified estimates of its parameters. In scenarios in which we lack sufficient observational data to constrain ESMs, we can often make use of statistical or prior information to make them identifiable. However, this often requires specifying nonphysical components for our model (i.e., shrinkage terms pushing some parameters to zero or priors enforcing spatial smoothness). With a lack of physical intuition to guide us in estimating what these empirical parameters may be, we may instead formulate this as a Bayesian inference problem concerning assimilation of data and priors with physical models. Once more, automatic differentiation becomes the focus because it is a key element of gradient-based Markov chain Monte Carlo methods, such as Hamiltonian Monte Carlo, that are at the heart of several widely used statistical frameworks (Carpenter et al., 2017; Salvatier et al., 2016). By using gradient-based Markov chain Monte Carlo for ESMs, we can potentially realize savings in time complexity in the number of model parameters from $O(N^2)$ to $\approx O\left(N^{\frac{5}{4}}\right)$ exhibited for other classes of statistical models (Mattingly et al., 2012; Pillai et al., 2012). This presents a tremendous new opportunity for applying a wide range of Bayesian modeling techniques to the study of Earth system science at scale. We can also realize a much simpler use of autodiff by recognizing that sensitivity analysis is largely a matter of computing derivatives of target variables with regard to inputs. Thus, we could potentially replace current methods that require a function space approximation in terms of a limited number of basis functions (Pianosi et al., 2016). These methods are used to circumvent computation of Hessian matrices, a task that autodiff is uniquely well suited to handle.

The greatest current barrier to making these breakthroughs is the time and effort required to translate our knowledge as encoded in existing community models into a format intelligible by automatic differentiation software. However, substantial progress has been made by the AI/ML community in developing transpilers capable of automatically mapping programs specified in a generic, high-level language into autodiff-enabled frameworks (Bradbury et al., 2018). This also aligns well with current pushes to use commercially available deep learning packages in high-performance computing environments administered by the US Department of Energy (Chien et al., 2019). We also posit that favoring model implementations in widely used open-source ML frameworks will aid in the communication and dissemination of related modeling work because it would allow us to build upon existing advances in model transfer, serialization, and dissemination that have already been made by the AI community.

References

Amos, B., and Kolter, J. Z. 2017. OptNet: Differentiable Optimization as a Layer in Neural Networks, in: International Conference on Machine Learning. Presented at the International Conference on Machine Learning, PMLR, pp. 136–145.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. 2018. JAX: composable transformations of Python+NumPy programs.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. 2017. Stan: A Probabilistic Programming Language. Journal of Statistical Software 76. https://doi.org/10.18637/jss.v076.i01

Chien, S. W. D., Markidis, S., Olshevsky, V., Bulatov, Y., Laure, E., and Vetter, J. S. 2019. TensorFlow Doing HPC. 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW) 509–518. https://doi.org/10.1109/IPDPSW.2019.00092

Hu, Y., Anderson, L., Li, T.-M., Sun, Q., Carr, N., Ragan-Kelley, J., and Durand, F. 2020. DiffTaiChi: Differential Programming for Physical Simulation 20.

Mattingly, J. C., Pillai, N. S., and Stuart, A. M. 2012. Diffusion limits of the random walk Metropolis algorithm in high dimensions. The Annals of Applied Probability 22, 881–930. https://doi.org/10.1214/10-AAP754

Pianosi, F., Beven, K., Freer, J., Hall, J. W., Rougier, J., Stephenson, D. B., and Wagener, T. 2016. Sensitivity analysis of environmental models: A systematic review with practical workflow. Environmental Modelling & Software 79, 214–232. https://doi.org/10.1016/j.envsoft.2016.02.008

Pillai, N. S., Stuart, A. M., and Thiéry, A. H. 2012. Optimal scaling and diffusion limits for the Langevin algorithm in high dimensions. The Annals of Applied Probability 22, 2320–2356. https://doi.org/10.1214/11-AAP828

Salvatier, J., Wiecki, T. V., and Fonnesbeck, C. 2016. Probabilistic programming in Python using PyMC3. PeerJ Computer Science 2, e55. https://doi.org/10.7717/peerj-cs.55

Schenck, C., and Fox, D. 2018. SPNets: Differentiable Fluid Dynamics for Deep Neural Networks. Arxiv 19.