Model Parameter Optimization: ML-guided trans-resolution tuning of physical models

Sam Partee
Cray, Inc.
Seattle, WA
spartee@cray.com

Michael Ringenburg
Cray, Inc.
Seattle, WA
mikeri@cray.com

Benjamin Robbins
Cray, Inc.
Seattle, WA
brobbins@cray.com

Andrew Shao
University of Victoria
Victoria, BC
aeshao@uvic.ca

Abstract

This paper proposes a novel method to automate and enhance the process of tuning physically-based numerical models. The Model Parameter Optimization (MPO) method learns from previous executions of numerical models to inform future executions. Specifically, MPO utilizes machine learning to construct a mapping between the free-parameter spaces of low-resolution models and high-resolution models, enabling users to tune their high-resolution models by executing much cheaper low-resolution models. We validate the MPO method by applying it to Modular Ocean Model 6 (MOM6)[7], and show that it achieves competitive tuning results.

1 Introduction

Large-scale, high-resolution numerical models are widely used in many scientific domains, including oceanography, high-energy physics, climate science, and material science. The models used often have tens or hundred of free parameters which provide input to parameterizations that approximate unresolved physical processes. For example, MOM6 has over 300 free parameters. Model users must tune these free parameters to optimize the skill of their models. However, tuning for high-resolution model runs is an expensive and time-consuming process that involves many executions of the model with different sets of parameters at the resolution of interest.

To address this problem, we propose a Model Parameter Optimization methodology that allows model users and developers to tune their physically-based numerical models at low-resolution, and then convert the tuned parameters to high-resolution equivalents. The process uses a pair of machine learning models to learn the mapping between the low-resolution parameter space and the high-resolution parameter space. First, the *Parameter Mapping Network* (or PMN) is trained to identify low-resolution parameters from low-resolution simulation data. Next, the *Parameter Mapping Transform* (or PMT) learns the relationship between the high-resolution and low-resolution parameter spaces by feeding high-resolution output to the PMN. Model users and developers can then apply the PMT to map parameters obtained from low-resolution tuning to parameters that can be used to execute the model at high-resolution.

In the remainder of this paper, we:

- describe the Model Parameter Optimization method in detail (Section 2);
- discuss the results of applying our method to the double-gyre test case of the MOM6 [7] ocean model (Section 3);
- survey related work on tuning of numerical models (Section 4);
- discuss our future plans for Model Parameter Optimization and other methods that apply machine learning and AI techniques to improve the skill and performance of scientific simulations (Section 5).

Second Workshop on Machine Learning and the Physical Sciences (NeurIPS 2019), Vancouver, Canada.

2 The Model Parameter Optimization Method

The research described here aims to decrease the amount of computational resources and developer time spent in tuning high resolution physical models. In order to accomplish this, we use machine learning to create a mapping between the parameter spaces of two different resolutions. The following sections detail stages required to apply MPO to any physical model.

2.1 Data Generation

Data generation is the first step of the MPO method. In order to construct a mapping between parameter spaces, a search space for the targeted tuning parameters must be created for both resolutions. The search space consists of a subset of all permutations within a sensible range of the desired tunable parameter space. Both resolutions are run with each tunable parameter space subset keeping the rest of the free parameters fixed.

2.2 Data Preprocessing

Simulation data requires a number of pre-processing steps before it can be used as training samples for a machine learning model. The output from many physical models, such as ocean models, are simulated on a multi-dimensional grid of points where values corresponding to the physical phenomena being simulated are captured. Collapsing this multi-dimensional grid of simulation data into a dataset for machine learning largely depends on the simulation model. The pre-processing method implemented in this research extracts simulation features at each grid point and combines them into a TFrecord dataset.

The same method of pre-processing is carried out on both the low and high resolution models. However, if pre-processed directly from the output data, model data at two different resolutions could not be used to infer from or train the same neural network as they would have different dimensions. To mitigate this, the Climate Data Operations(CDO) tool[11], developed by the Max Planck Institute for Meteorology, is used in this research to conduct a conservative remapping of grids between resolutions.

2.3 Parameter Mapping Network

After the simulation data has been processed, the PMN is trained to predict the selected tunable parameters from a low resolution simulation output. The training process is supervised where the targets are the selected tunable parameters. The loss function that guides the training is the mean squared error between the predicted values of the tunable parameters and the actual tunable parameters used in the low-resolution run.

2.4 Parameter Mapping Transform

After a suitable accuracy is met by the PMN, runs of higher resolution are given as samples to the PMN without adjusting the weights of the network for error. Giving the PMN a higher resolution sample after it has been trained to predict from low-resolution inputs provides the mapping space. The mapping space is the tunable parameter space of the high resolution simulation as if it were a low-resolution simulation. Since the PMN has no knowledge that the resolution has been increased, when the PMN is given a sample of higher resolution, it predicts the low-resolution tunable parameters of a low resolution simulation most similar to the high resolution simulation. The difference between the mapping space and the actual high resolution parameter space constitutes the effect of increasing the simulation resolution.

The Parameter Mapping Transform(PMT) distills the knowledge from the PMN by regressing between the high resolution parameter space and the mapping space across multiple simulations. The knowledge captured in the PMT can be interpreted as an approximation of the non-linear, multi-objective function between the tunable parameter spaces of a model at two different resolutions. A brute force mapping of this approximation might compare every permutation of all tunable parameter values, but that approach would be infeasible on practically any numerical model requiring thousands of computationally expensive simulations. In this research, we show that our MPO method can approximate this non-linear mapping with a fraction of the runs of that of the brute force algorithm.

3 Results

To evaluate our method, we used Modular Ocean Model 6 (MOM6) [7] to simulate an idealized double-gyre case that mimics the behavior of the North Atlantic Ocean. The relative simplicity of this model makes it an ideal test, as it can be run relatively cheaply even at high resolutions. We ran MOM6 at the 20km and 40km resolutions, and targeted two tunable parameters, eddy viscosity (KH) and thickness diffusion (KHTH). We held the rest of the free parameters constant.

3.1 Training Data

We ran 283 configurations of low resolution (40km) and high resolution (20km) during the data generation stage. The low and high were run for 100 and 30 years respectively. A conservative first 20 years of simulation is discarded due to model spinup. For training, each simulation was split into 5 year averages for a single sample which means 16 samples are taken from each model run of 100 years. For training, there were a total of 4,528 low resolution samples(283×16). The features in each sample were one or more physical quantities captured at each grid point. These features include kinetic energy (KE), stream function value (STR), and sea surface height (SSH).

3.2 Machine learning models

The Parameter Mapping Network (PMN) is a Deep Neural Network (DNN). The DNN uses rectified linear unit (ReLU) activation and the Adam optimizer [5]. The network ends in a dense layer with two outputs for the two parameters being mapped.

The Parameter Mapping Transform (PMT) is a gradient boosting algorithm known as XGBoost [3]. Typically an XGBoost regression model is restricted to a single target variable, however, the model can be used to predict multiple target variables by wrapping it in a MultiOutputRegressor from sklearn [8]. Wrapping the XGBoost model in this fashion allows for the prediction of n variables in the tunable parameter space. In our experiments, we used two tunable parameters (KH and KHTH).

3.3 Results

All datasets and models included in the results were able to reach a test accuracy of over 95% for predicting the low resolution parameter space of a given low resolution model output. Through hyper-parameter optimization using the Cray AI HPO package [4], a DNN trained on the KE-only dataset recorded the lowest error with a mean absolute error of 31.1.

After the PMN was trained, processed runs of high resolution were given as samples to infer the mapping spaces. There are a total of 283 (total runs) $\times$ 2 (number of samples generated from each run) high resolution samples. These 566 samples provide the PMT with the mapping spaces. By comparing these mapping space values to the actual high resolution tunable parameter space across multiple runs, the non-linear mapping between the parameter spaces is learned by the PMT. Once trained to infer high resolution tunable parameters from low resolution parameters, the PMT can be used as an oracle for tuning the selected parameters.

To verify our mapping we implemented a brute force heuristic we call the Automatic Tuning Algorithm (ATA). We implement the ATA as a method to benchmark performance as the novelty of our method objective makes it difficult to compare with previous automated tuning approaches. The ATA utilizes high resolution data (5km) to calculate the proper value for a metric known as jet penetration. The jet penetration calculation is shown in the appendix in figure 2. The average jet penetration is taken, discarding the first ten years, for each of the 40km, 20km and 5km models. The parameter spaces for each resolution chosen by the ATA are the ones that produce the simulations resulting in the jet penetration average closet to the 5km resolution simulation value.

The ATA found that the optimal tunable parameters for the low resolution model (40km) were a KH of 2250 and a KHTH of 1750. For the 20km model, the optimal values of the tunable parameters were a KH of 750 and and a KHTH of 2000. Therefore, if the mapping generated by the MPO method is correct, it should produce KH and KHTH values close to the high resolution (20km) parameter space of (750, 2000) when given the low resolution (40km) parameter space of (2250, 1750). As shown in table 1, the predictions given by MPO generated mapping are very close to that of the ATA. The

Model	Dataset	Opt KH	Opt KHTH	Pred KH	Pred KHTH	JP % Difference
DNN-1	KE	750	2000	1013	2164	0.7322%
DNN-2	KE,STR,SSH	750	2000	1219	1507	0.6201%
DNN-3	KE	750	2000	922	2056	0.3178%

Table 1: The results of three model predictions for the predicted high resolution parameter space and the optimal high resolution parameter space chosen by the ATA, as well as the percent difference between the jet penetration values when run with the optimal and predicted parameters.

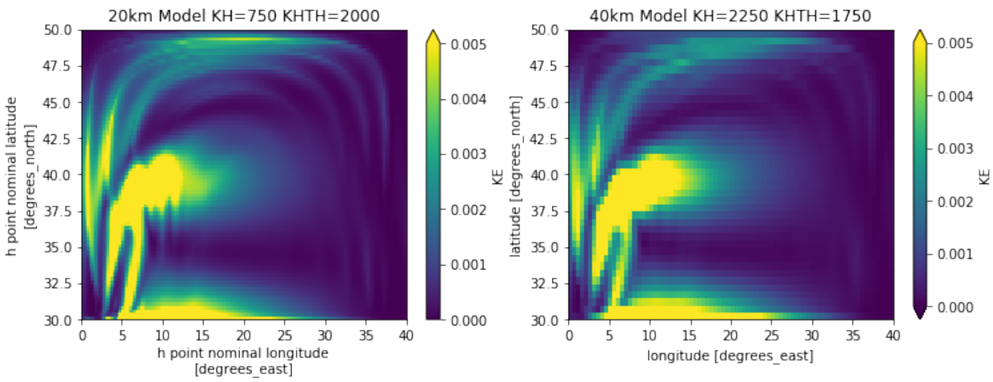

Figure 1: This figure shows the two pcolormesh graphs of kinetic energy for the parameter space values chosen to be optimal by the ATA for the 40km and 20km models.

DNN-3 model comes the closest of any model with the predicted high resolution tunable parameter space of KH=922, KHTH=2056.

The goal for tuning the 20km double-gyre model with MPO was to get within the .3185 standard deviation of the 5km model jet penetration. In other words, MPO tuned, 20km models should have less than a 1.8% difference in jet penetration compared to the 5km model. As 2 shows, all MPO tuned models had less than a 1% difference in jet penetration from the 5km model.

4 Related Work

Current tuning processes are often undocumented and performed manually by model experts. Schmidt et al. [10] surveys the techniques used at six large US-based centers. Many attempts at automated parameter tuning have run into issues with overfitting (as will be familiar to anyone who has worked with ML models), however a few approaches such as Zhang et al. [14] and Severijns et al. [13] have demonstrated improved numerical model skill through automated optimization techniques. Our Model Parameter Optimization algorithm is orthogonal to automated tuning approaches. We instead focus on how tuned results can be transferred between resolutions, thereby significantly reducing the cost of both manual and automated tuning techniques.

An alternative to tuning numerical parameters is replacing the parameterizations they control with ML-based models [2, 1, 9, 6, 12]. Machine learning models can in some cases offer an effective replacement for traditional numerical parameterizations. However, numerical parameterizations are still widely used and offer better explainability. We expect future models to use a combination of the two, and thus still require numerical tuning.

5 Conclusions and Future Work

This paper described a method for tuning parameters of a high-resolution physically-based numerical model using a low-resolution version of the numerical model and a pair of machine learning models. We have shown that our method is effective for tuning the MOM6 ocean model, and allows scientists to save significant effort, time, and computational expenditure. To improve our method, we are working on a second version of MPO that will make the method more generalizable and address current limitations such as initial dataset size.

Our research into using AI to enhance and improve physically-based numerical model continues. We are working with new models across domains to further validate the Model Parameter Optimization methodology. We are also investigating new methods and creating tools to facilitate the integration of machine learning techniques with numerical simulations.

References

[1] Thomas Bolton and Laure Zanna. Applications of deep learning to ocean data inference and subgrid parameterization. *Journal of Advances in Modeling Earth Systems*, 11(1):376–399, 2019.

[2] Noah D Brenowitz and Christopher S Bretherton. Prognostic validation of a neural network unified physics parameterization. *Geophysical Research Letters*, 45(12):6289–6298, 2018.

[3] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 785–794, 2016.

[4] Cray Inc. Cray-AI Hyperparameter Optimization. `https://pubs.cray.com/content/S-2589/1.2.UP00/xctm-series-urika-xc-analytic-applications-guide/hyperparameter-optimization-hpo-support`.

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[6] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.

[7] Mom6: Modular Ocean Model 6. `https://github.com/NOAA-GFDL/MOM6`.

[8] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vander-Plas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830, 2011.

[9] Stephan Rasp, Michael S Pritchard, and Pierre Gentine. Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, 115(39):9684–9689, 2018.

[10] G. A. Schmidt, D. Bader, L. J. Donner, G. S. Elsaesser, J.-C. Golaz, C. Hannay, A. Molod, R. B. Neale, and S. Saha. Practice and philosophy of climate model tuning across six us modeling centers. *Geoscientific Model Development*, 10(9):3207–3223, 2017.

[11] Uwe Schulzweida. Cdo user guide. `https://doi.org/10.5281/zenodo.2558193`, February 2019.

[12] Ole Schütt and Joost VandeVondele. Machine learning adaptive basis sets for efficient large scale density functional theory simulation. *Journal of chemical theory and computation*, 14(8):4168–4175, 2018.

[13] CA Severijns and W Hazeleger. Optimizing parameters in an atmospheric general circulation model. *Journal of climate*, 18(17):3527–3535, 2005.

[14] T Zhang, L Li, Y Lin, W Xue, F Xie, H Xu, and X Huang. An automatic and effective parameter optimization method for model tuning. *Geoscientific Model Development*, 8(11):3579–3591, 2015.

Appendix

5.1 Result Verification

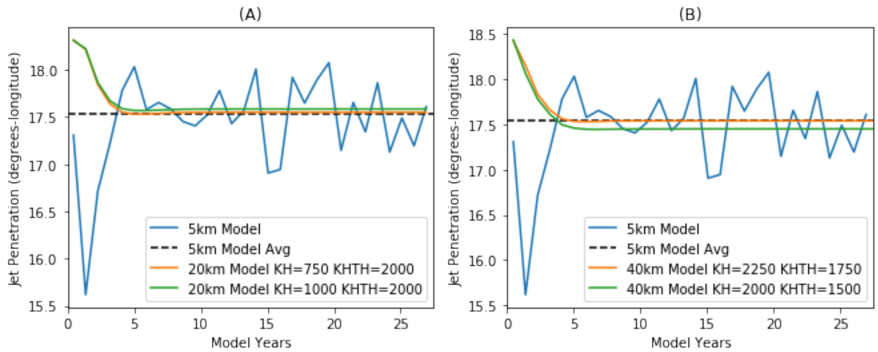

Figure 2: A shows the jet penetration metric plotted over the first 30 years of simulation for the 40km model. B shows the same for the 20km model. The 5km jet penetration average is shown as the dashed black line.

To understand the results of the parameter mapping, it is crucial to understand the physical effects of the viscosity (KH) and the thickness diffusivity (KHTH). In the double-gyre test case there are two main energy sources for the turbulent field. One is the large-scale velocity field where horizontal velocity shear can generate eddies. The other is the sloping of isopycnals which can generate baroclinic turbulence. Viscosity, as discussed previously, can reduce velocity gradients and dissipates kinetic energy to fricative effects. Thickness diffusivity acts to reduce horizontal density gradients, reducing the available energy to generate baroclinic turbulence. Higher values of both KH and KHTH thus serve to reduce the total energy available to the energize the turbulent field, albeit in different ways. However in general, there are potentially multiple combinations of KH and KHTH that yield similar large-scale features. Because the model of higher resolution(the 20km model) resolves higher amounts of turbulence by nature, the ATA results are logical to decrease KH(from 2250 to 750). The 20km model should have less KH than the 40km model because in the case of the 40km model some of the large-scale reduction of velocity field is missing because it has a weak turbulent field. In the case of the the 20km model, the resolved turbulent field is effectively taking the place of the higher KH.

5.2 Machine Learning Models

Model Name	Layers	Optimizer	Activation	Features	Dataset Size
DNN-1	4	Adam	ReLU	KE	4528*64000
DNN-2	4	Adam	ReLU	KE,STR,SSH	4528*89600
DNN-3	6	Adam	ReLU	KE	4528*64000

Table 2: This table details the models and datasets included in the results of the MPO application to the MOM6 double-gyre model