Regression methods as emulators

Ayush Prasad, Kwabena Atobra, Topi Laanti

December 23, 2022

1 Introduction

Mathematical models that describe various phenomena across a multitude of different fields of science have enabled researchers to predict and understand what different features contributes to the phenomena they study [7]. These complex models include and implement fully non-linear ordinary or partial differential equations, multivariate latent factor models and agent-based models, often combined together. These mathematical models, often called simulations or **simulators**, enable researchers to numerically simulate phenomena that are not easily observed due to practical, economical or ethical reasons.

However, in order for the mathematical models to provide insight and new knowledge, they must be precise and provide accurate results. The simulators that are accurate are often computationally demanding, meaning that it can take a lot of resources and time to get the required results. This means that this computation often can not be created in an environment that does not have the computational resources required or if the simulation results are needed in a short time-span. The computational cost also limits simulators in more extensive parameter exploration, and when data size are massive in scale, and in uncertainty quantification [3].

Speeding up simulators enable them to provide results that are too prohibitively expensive and make progress on scientific research faster. The speedup in simulators often come as a tradeoff in accuracy, and thus it is important to maintain a reasonable level in both to get best results, and finding methods that do this is an active are of current research.

A popular approach for speeding up simulators is to use machine learning models as **emulators** [3]. In this context, emulators are machine learning models that try to reproduce the output of simulators. These machine learning models do not fully replace the simulators, as at the very least, simulator outputs are needed to train the machine learning models in order for them to produce accurate outputs. Depending on the use case, they can be used in tangent with simulators, replace the emulators after the initial tuning, or in a multi-fidelity approach where both simulators and emulators are used in conjunction to provide optimal results that rely on the more accurate simulators and more fast emulators. These emulators are often machine learning regression models.

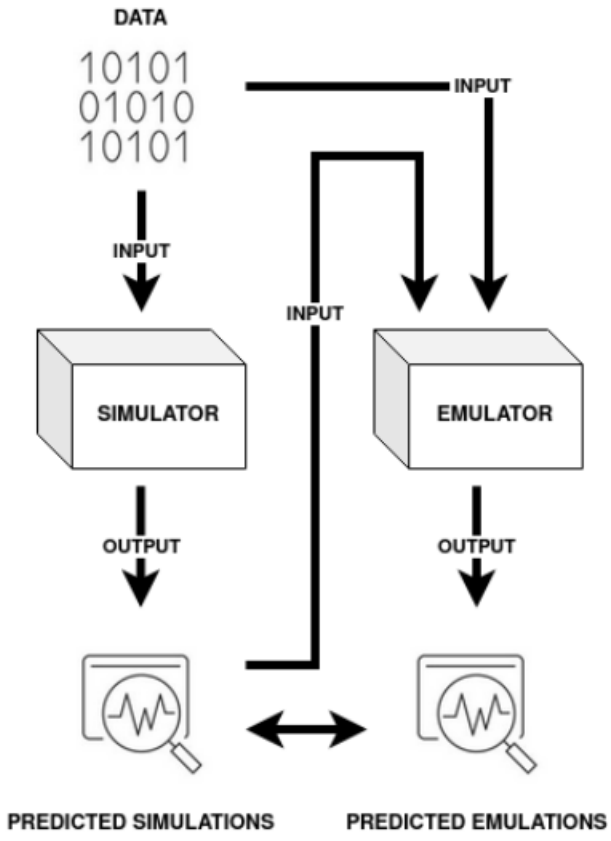


Figure 1: Simulator and Emulator pipeline.

2 Regression Methods

Machine learning is used in various fields of science to predict and gain insight on problems that cannot be solved using traditional computational or statistical approaches [4]. Regression is a subfield of supervised machine learning, a type of machine learning where labeled datasets are used to predict continous outcomes. Regression is used to forecast and predict certain outcomes and also find causal relationship between the independent and dependent variables. However, regression alone only shows the relations between a dependent variable and a dataset of different variables, meaning if some relations are not seen on the dataset used, regression does not reflect these causal relationships.

Regression analysis predicts the dependent variable y for independent variables $\mathbf{x}$. At its simplest, a simple linear regression model defines the dependence of variable $y = \beta_0 + \beta_1 x_1 + ...\beta_n x_n + \epsilon$. While linear regression is very easy to understand and the causal relationship between the weights $beta_i$ and the independent variables x make it easy to interpret, it can not model complicated non-linear relationships between the variables. To this end, many more complicated models have been developed, that provide more accurate predictions at the cost of less being less interpretable. Models that are common in the realm of emulators are random forests, Gaussian processes, and various neural network deep learning algorithms [3].

3 Practical examples

Emulators have become vital in speeding up real-time simulations of complex processes such as those occurring in geoscience [2], computational medicine [6], ecology etc.

3.1 Geophysical, Atmospheric and Planetary sciences

While computer simulations are very useful tools for scientific discoveries in geophysics, atmospheric, astrophysics and high energy physics where usually direct observations or designing experiments are very expensive or mostly impossible. Numerical simulations have traditionally been used in these fields to understand complex processes occurring in these fields. As mentioned in the first section, executing accurate simulations are often too slow which limits their applicability in uncertainty quantification, large scale data analysis and exploring parameters [3]. Emulators present a promising route to accelerate simulations using machine learning and deep learning techniques. This allows accurate results that are comparable to those obtained from numerical simulations to be obtained with only a fraction of the resources and time taken. In this section some examples of emulators used in geophysics, atmospheric and planetary sciences are thoroughly discussed.

3.1.1 Emulators in Geophysics

Emulators have been used to study continuum scale brittle fracture simulations [2] and seismic tomography studies [2] among others. **Brittle failure** under dynamic loading results in the propagation and coalescence of microcracks which is very computationally expensive or sometimes intractable to simulate. Accelerating high-strain continuum scale brittle fracture simulations using machine learning emulators have led to about four order of magnitudes as well as a dimensionality reduction of the problem. Using the just the length of the longest crack and one of the maximum stress components to capture the necessary physics [2]. Shock wave physics is studied by impacting a target plate with a flyer plate to induce mode I crack growth which induces non-homogeneous damage distribution in the target plate [2].

The physical problem is studied with an experimental set up while the numerical simulations is carried out by the Hybrid Optimization Software Suite (HOSS). Other simulatation tools such as FLAG are much faster that HOSS but performs poorly after 0.8 micro seconds since it neglects the effects of damage evolution of the such wave physics. Coupling FLAG and HOSS slightly improves the simulation speed, however it still suffers from the drawbacks of HOSS beyond 0.8 micro seconds as illustrated in figure 1. A machine learning emulator is used to replace the computationally expensive HOSS simulator.

The machine learning model is an LSTM model that takes the longest crack length over time and the maximum stress over time as input parameters. The output is the shock wave velocity over time. The data set consisted of 100 simulations with a train test split (70-30 percent). The model architecture is illustrated in figure 2. How the model performs on the test set and an out of sample validation set are illustrated in figure 3.

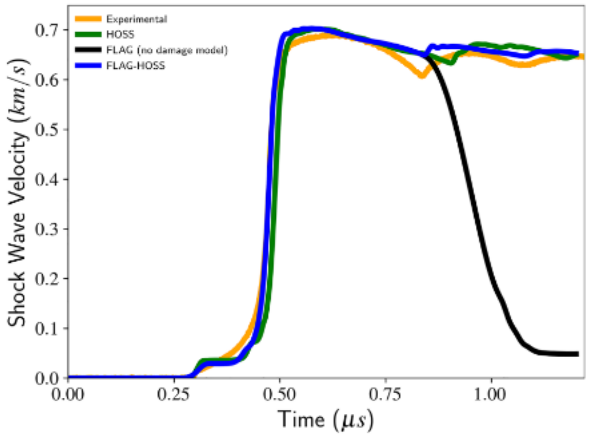


Figure 2: Result of high fidelity simulators compared to experimental output [2]

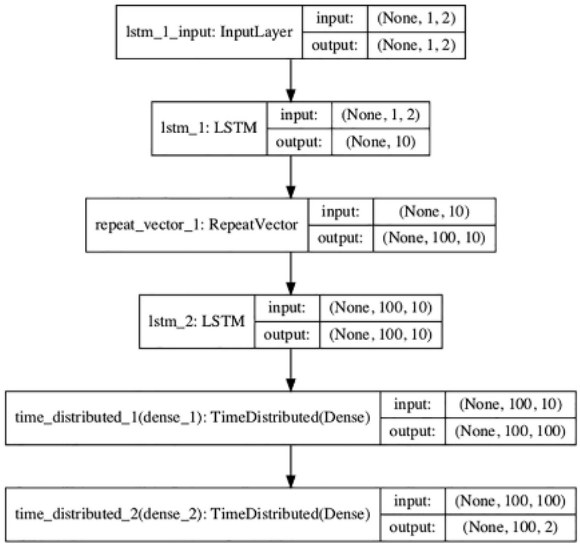
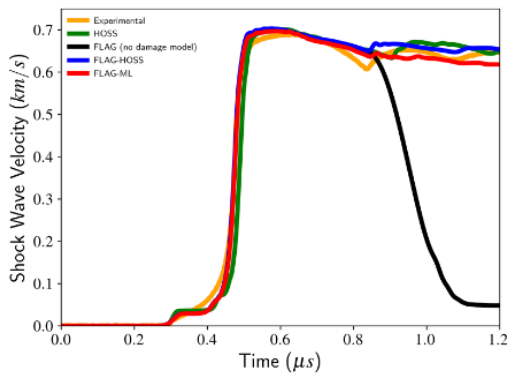


Figure 3: FLAG-ML model architecture [2]

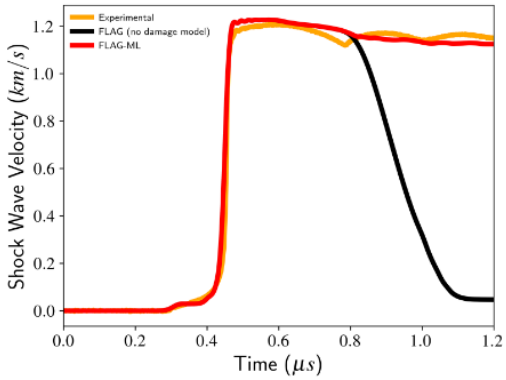
Seismic tomographic inversion of the Shatsky Rise oceanic plateau has been carried out using a Deep Neural Network Search (DENSE) as an emulator. Using the initial velocity profile and regularization in the optimization as input parameters, the velocity structure and crustal thickness as a function of position in the Shatsy Ridge that matches the seismic reflection data is inverted for [3]. Without using an emulator, uncertainty quantification would be prohibitively expensive as the simulator would have to be executed multiple times [3].

3.1.2 Emulators in Atmospheric sciences

In the field of atmospheric sciences, ECHAM-HAM which is a high fidelity simulator for global climate aerosol modeling is used to calculate the distribution and evolution of both internal and external distribution of aerosol species in the atmosphere and how they affect radiation and cloud processes. The simulators inputs include the scaling of the emissions



(a) FLAG-ML test results [2]



(b) FLAG-ML validation results [2]

Figure 4: FLAG-ML emulation results

flux of Black Carbon, scaling on the removal rate of Black Carbon through wet deposition and scaling of the imaginary refractive index of Black Carbon between 0.2 and 0.8 [3]. The model output is the aerosol absorption optical depth. The model is computationally expensive and can require several CPU hours. DENSE has been used to successfully emulate the simulator even using few training data points.

3.1.3 Emulators in Planetary sciences

Halomod is a high fidelity simulator for galaxy Halo modelling. It is used in calculating calculating correlation functions, angular scales, redshift, and cosmological models. Emulators are extremely important for Galaxy Halo modelling as researchers are used interested in extracting parameters for multiple and different galaxy populations. This fast parameter retrieval is very difficult when working with slow high fidelity simulators prompting the need for machine learning emulators [3].

3.1.4 Emulators in Ecology

In ecology, process-based models are used to understand the mechanisms of vegetation growth and land-atmosphere interactions and predict properties such as Leaf Area Index (LAI), Net Ecosystem Exchange (NEE), soil moisture, etc. These models that represent our understanding of the physical processes have a high number of parameters that need to be calibrated for each location before a model is used. The calibration process is time intensive as the models are slow to run and a high number of parameter combinations have to be tried out. To make this process faster, the authors in [1] trained a gaussian process emulator on the outputs of the SIPNET model to predict ecosystem variables such as NEE. The emulator was then used inplace of the actual SIPNET model in the calibration process. The parameters obtained by using the emulator had similar performance as the original SIPNET model and reduced the computation time by half.

3.1.5 Other Application domains

Machine learning emulators have been used in other and often diverse fields in the natural sciences such as studying ocean biogeochemistry models, inertial confinement fusion, Optical Thomson scattering, X-ray emission spectroscopy etc [3]. Emulators have also been used to provide near-real simulations of cardiac electromechanics achieving a massive improvement in computational time. This is shown in Figure 4.

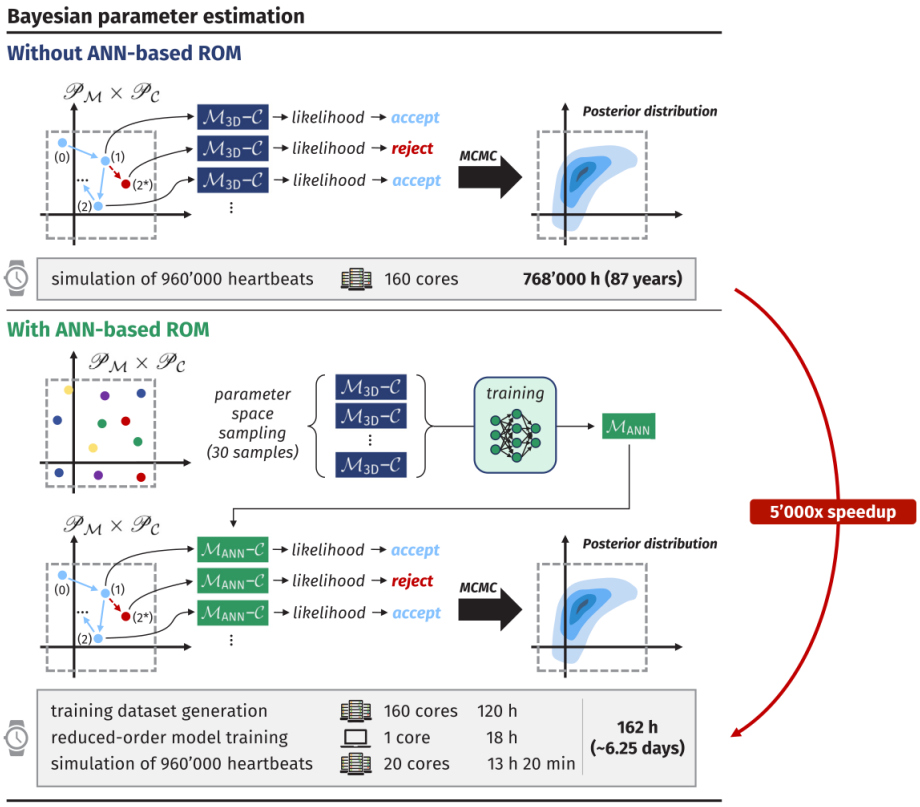


Figure 5: Comparing emulator to high fidelity simulator for cardiac electromechanics [6]

3.1.6 Incorporating physical constraints in emulators

In a typical data-driven emulator, the emulator is trained on the outputs of the original model, however, when the size of training samples is low it can be hard to generalize the model. To increase the accuracy and reliability of the model in such scenarios, prior knowledge about the scientific phenomenon being studied can be incorporated into the model. Towards this goal, the authors in [5] tried to improve the predictions of lake water temperature produced by the General Lake Model (GLM) by using a physics-guided machine learning approach. The relationship between the temperature and density of water in a lake can be explained by a known empirical equation. This relationship can be included as an additional loss term in a recurrent neural network. The accuracy of this hybrid model is higher than that of using the RNN or the physics-based model (GLM) independently.

3.1.7 Towards a general purpose emulator

Most emulators like most machine learning models are developed for a specific use case and can not be extended to other cases without a modification in model architecture and training procedure. However, new methods have been used to build emulators that approximate a general purpose emulator, in the sense that it can be used to emulate models from different and diverse domains without modification to the internal architecture of the emulator. This approach is known as Deep Neural Network Search (DENSE) [3]. At its core DENSE is a convolutional neural network (CNN) with a super neural network architecture, that allows the right network architecture to found during the training phase of the model. In some sense, the model architecture is treated as another hyper-parameter. The model architecture is shown in figure 5.

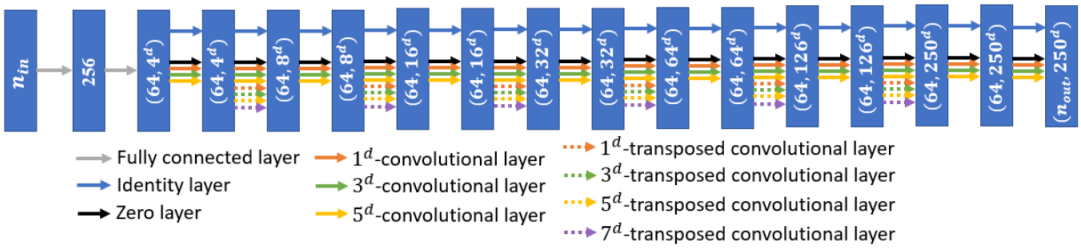
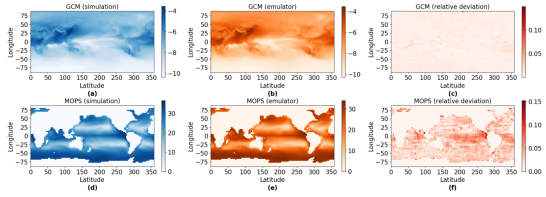


Figure 6: DENSE super-architecture [3]

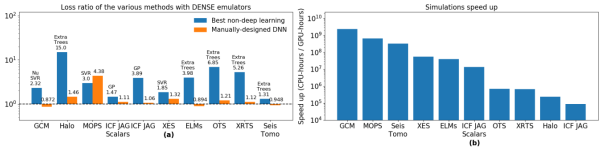
CNNs generally have good priors on natural signals as opposed to other ML models such as random forests and gaussian processes which make them a natural choice for building models in the natural sciences. A general purpose emulator which is fast and accurate would allow for rapid testing of ideas and near real-time prediction based experiment control and optimization [3]. Results of using DENSE to emulate high fidelity simulators for global climate models (GCM), large scale ocean biogeochemistry models (MOPS) etc, are shown in Figure 6.

3.1.8 Limitations of Emulators

Emulators like most ML models are data hungry and require data to be drawn from the high fidelity simulator to train and test the emulator. This imposes a natural disadvantage in constructing emulators especially when the simulator is extremely slow since the data



(a) Results for GCM and MOPS emulation using DENSE [3]



(b) Speedup and Loss result from DENSE [3]

Figure 7: Results obtained from different domains using DENSE

generation process would be prohibitively slow. Hence robust emulators are cleverly built to produce accurate results even with a limited amount of data. However, the benefit of training and deploying emulators even when dealing with slow emulators usually far exceed using only simulators without emulators.

On the performance side, emulators may struggle when dealing with models that have regions of high variability. That is when slightly varying input parameters leads a high variation in the output in those regions. This limitation is also observed in other deep learning models since regions with high variability need more data to adequately probe as compared to regions with low variability [3].

4 Conclusion

Emulators have a wide range of applications they can provide, while providing the benefits of large speedups that provide researchers and applications to do the computation required in much more feasible time. Emulators have the tradeoff usually having a worse accuracy than the simulators they are based on, but depending on the application this may be a minor issue when you consider the benefits of large speedups that they provide.

Emulators will more likely become more commonplace as applications increase, as the real world data sets are becoming too massive in scale to compute using traditional simulations.

References

[1] I. Fer, R. Kelly, P. R. Moorcroft, A. D. Richardson, E. M. Cowdery, and M. C. Dietze. Linking big models to big data: efficient ecosystem model calibration through bayesian model emulation. *Biogeosciences*, 15(19):5801–5830, 2018.

[2] M. Fernandez-Godino, N. Panda, O. Daniel, L. Kevin, A. Hunter, R. Haftka, and G. Srinivasan. Accelerating high strain continuum-scale brittle fracture simulations with machine learning. *Computational Materials Science*, 186(1):0927–0256, 2020.

[3] M. Kasim, D. Watson-Parris, L. Deaconu, S. Oliver, P. Hatfield, D. Froula, G. Gregori, M. Jarvis, S. Khatiwala, J. Korenaga, et al. Building high accuracy emulators for scientific simulations with deep neural architecture search. *Machine Learning: Science and Technology*, 3(1):015013, 2021.

[4] D. Maulud and A. M. Abdulazeez. A review on linear regression comprehensive in machine learning. *Journal of Applied Science and Technology Trends*, 1(4):140–147, 2020.

[5] J. S. Read, X. Jia, J. Willard, A. P. Appling, J. A. Zwart, S. K. Oliver, A. Karpatne, G. J. A. Hansen, P. C. Hanson, W. Watkins, M. Steinbach, and V. Kumar. Process-guided deep learning predictions of lake water temperature. *Water Resources Research*, 55(11):9173–9190, 2019.

[6] F. Regazzoni, M. Salvador, L. Dede, and A. Quarteroni. A machine learning method for real-time numerical simulations of cardiac electromechanics. *Computer methods in applied mechanics and engineering*, 393(114825), 2022.

[7] P. Stolfi and F. Castiglione. Emulating complex simulations by machine learning methods. *BMC bioinformatics*, 22(14):1–14, 2021.