Physics Informed Spatiotemporal Deep Learning

Author: Aleksandr Mkrtchyan
BS in Data Science
AUA

Author: Jon Hakobyan
BS in Data Science
AUA

Supervisor: Aleksandr Hayrapetyan
Yerevan State University

***Abstract*—This capstone thesis presents a comprehensive investigation of physics-informed spatiotemporal deep learning, a novel approach for solving nonlinear partial differential equations (PDEs) with the aid of deep learning techniques. By leveraging the principles of physics-based deep learning, this research aims to provide accurate, data-driven solutions and data-driven discoveries to complex PDE problems. In particular, we discuss Burgers and the complex Ginsburg-Landau equations in the context of physics-informed neural networks.**

***Index Terms*—Keywords: physics-informed neural networks, spatiotemporal deep learning, nonlinear partial differential equations, physics-based deep learning**

I. Introduction

The surge in accessible data and computational capabilities has led to remarkable progress in machine learning and data analysis across diverse scientific domains, such as image recognition, cognitive science, and genomics. Nonetheless, data collection costs can be daunting when examining complex physical, biological, or engineering systems, often leading to insufficient information. In these cases, most cutting-edge machine learning methods, including deep, convolutional, and recurrent neural networks, are not robust and lack convergence guarantees.

It might seem improbable to train a deep learning algorithm to precisely identify a nonlinear map from high-dimensional input and output data pairs with only a few examples. Yet, prior knowledge, like governing physical laws, established rules, or domain expertise, can serve as a regularization agent to restrict the solution space. Integrating this structured information into a learning algorithm augments the data's informational value, enabling the algorithm to swiftly identify the right solution and generalize effectively, even with limited training examples.

Recent research [1]–[3] has highlighted the possibility of using structured prior information to develop data-efficient, physics-informed learning machines. These studies employed Gaussian process regression [4] to create functional representations customized for particular linear operators, successfully deducing solutions and offering uncertainty estimates for several prototype problems in mathematical physics. Although extensions to nonlinear problems [5] were proposed in subsequent studies, they encountered limitations: the necessity to locally linearize nonlinear terms in time and the Bayesian nature of Gaussian process regression demanding prior assumptions that could restrict the model's representational capacity and cause robustness/brittleness concerns, especially in nonlinear problems.

II. Methodology

In this study, we adopt an alternative approach by utilizing deep neural networks, capitalizing on their renowned ability as universal function approximators [6]. This allows us to directly address nonlinear problems without the need for prior assumptions, linearization, or local time-stepping. We take advantage of recent advances in automatic differentiation [7] to differentiate neural networks with respect to their input coordinates and model parameters, resulting in physics-informed neural networks (PINNs). These networks adhere to any symmetries, invariances, or conservation principles derived from the physical laws governing the observed data, as modeled by general time-dependent and nonlinear partial differential equations (PDEs). This straightforward yet effective framework enables us to address a wide array of problems in computational science and paves the way for potentially groundbreaking technology in the development of new data-efficient and physics-informed learning machines, novel numerical solvers for PDEs, and innovative data-driven approaches for model inversion and system identification.

The overarching goal of this work is to establish the groundwork for a new paradigm in modeling and computation that combines deep learning with long-established developments in mathematical physics. Our manuscript is divided into two parts, focusing on two key problem classes: data-driven solutions and data-driven discovery of PDEs. All code and data sets accompanying this manuscript can be found on GitHub. We employ relatively simple deep feed-forward neural network architectures with hyperbolic tangent activation functions and no additional regularization. Each numerical example in the manuscript includes a detailed discussion of the used neural network architecture and training process (e.g., optimizer, learning rates, etc.). We consider parametrized and nonlinear PDEs of the general form

$$u_t + N[u;\lambda] = 0,\ x \in \Omega,\ t \in [0,T],$$

where $u(t,x)$ denotes the latent (hidden) solution, $N[\cdot;\lambda]$ is a nonlinear operator parametrized by λ, and Ω is a subset of R^D. This setup encapsulates a wide range of problems in mathematical physics, including conservation laws, diffusion processes, advection-diffusion-reaction systems, and kinetic equations. For instance, the one-dimensional Burgers equation [8] serves as a motivating example, where

$$N[u,\lambda] = \lambda_1 u u_x - \lambda_2 u_{xx} \text{ and } \lambda = (\lambda_1, \lambda_2).$$

Here, the subscripts denote partial differentiation in either time or space. Given noisy measurements of the system, we aim to address two distinct problems. The first is inference, filtering and smoothing, or data-driven solutions of PDEs, which asks: given fixed model parameters λ, what can be said about the unknown hidden state $u(t, x)$ of the system? The second problem is learning, system identification, or data-driven discovery of PDEs, which seeks to determine the parameters that best describe the observed data.

III. Results and Discussions

A. Burgers Equation's Implementation Through TenserFlow

Utilizing the Chebfun package in conjunction with traditional spectral techniques, a high-resolution dataset was derived from the Burgers equation. Commencing with an initial condition $h(0, x) = 2\,\mathrm{sech}(x)$ and adopting periodic boundary constraints $h(t, -5) = h(t, 5)$ and $h_x(t, -5) = h_x(t, 5)$, the PDE was integrated until $t = \frac{\pi}{2}$. A spectral Fourier discretization consisting of 256 modes was employed, alongside a fourth-order explicit Runge-Kutta temporal integrator with a time-step of $\partial t = \frac{\pi}{2} * 10^{-6}$, ensuring precise and accurate results.

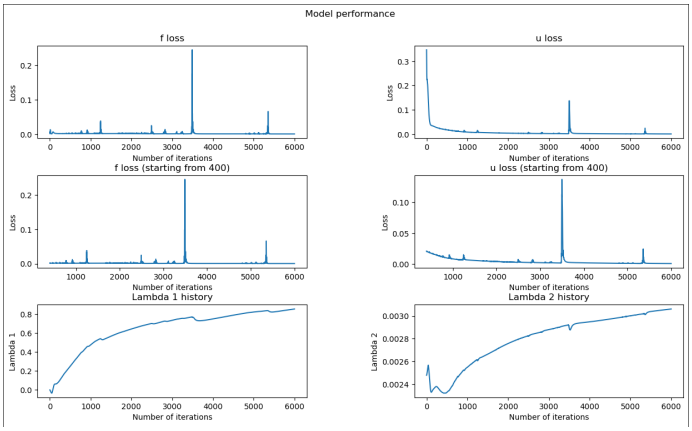

Fig. 1. Model performance: Noiseless data.

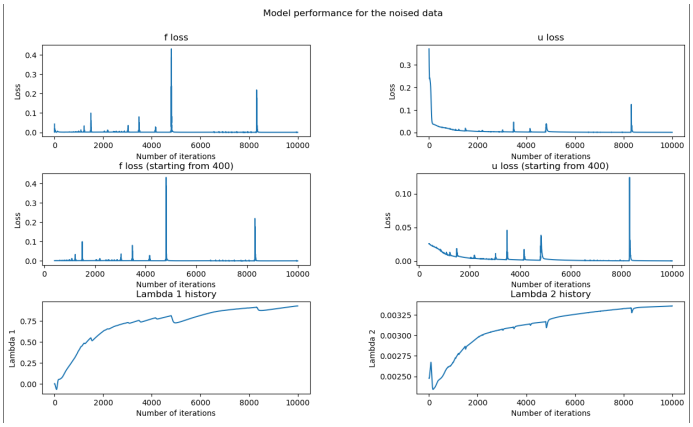

Fig. 2. Model performance: Noised data.

B. Burgers Equation's Implementation Through Pytorch

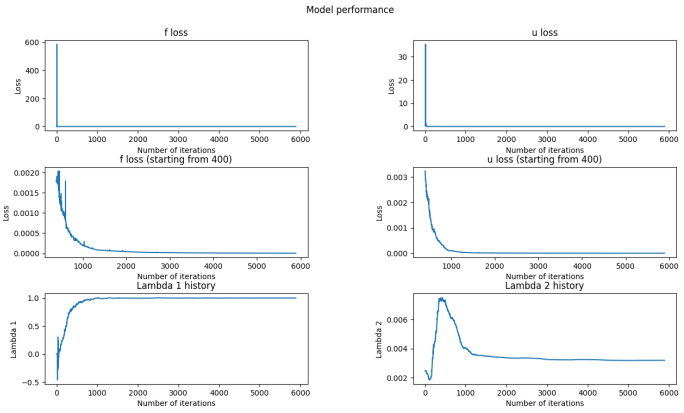

Fig. 3. Model performance: Noiseless data.

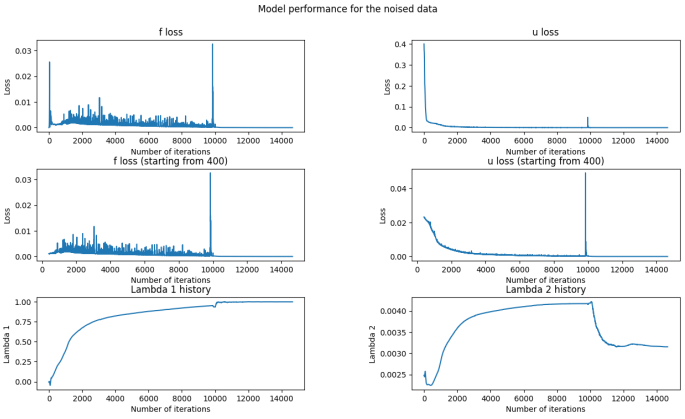

Fig. 4. Model performance: Noised data.

C. Similarities and Differences

Analyzing the outcomes, it becomes evident that the PyTorch implementation of Physics-Informed Neural Networks (PINNs) exhibits greater adaptability and superiority compared to its TensorFlow counterpart. Moreover, the PyTorch version demonstrates enhanced performance characteristics. Nonetheless, TensorFlow's training duration is considerably lengthened, albeit with reduced memory consumption. PyTorch facilitates rapid prototyping, while TensorFlow presents a more viable alternative for incorporating bespoke features within the neural network architecture. Notably, PyTorch now also supports this functionality. The PyTorch-based solution exhibits improved stability. In the PyTorch scenario, we employed the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) optimizer for minimizing the loss function, while the Adaptive Moment Estimation (ADAM) algorithm was utilized for optimizing the state loss. Conversely, TensorFlow solely relied on the ADAM algorithm for both purposes, as its LBFGS implementation resulted in memory overflow. Consequently, the PyTorch implementation emerges as the superior performer.

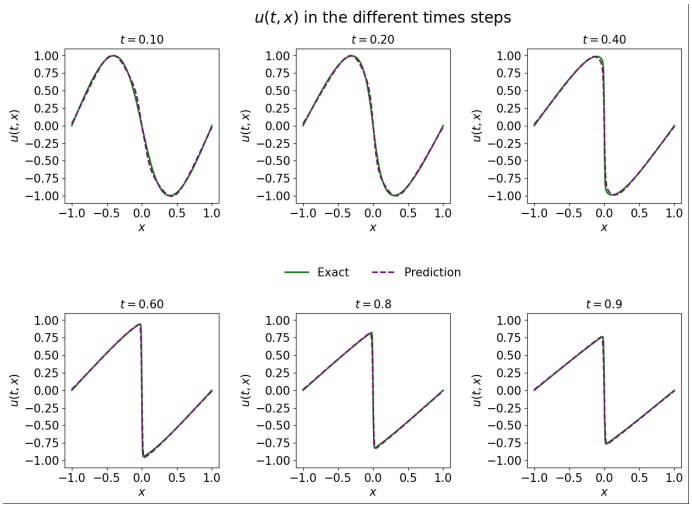

Fig. 5. Latent Solution on Tensorflow.

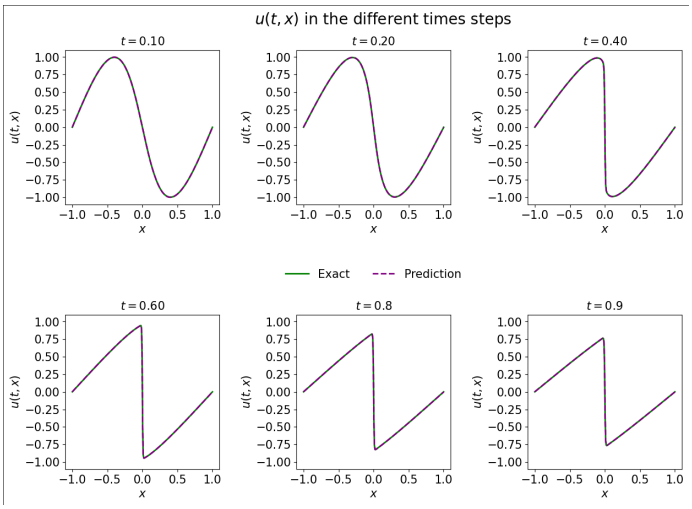

Fig. 6. Latent Solution on PyTorch.

Having the same number of iterations performed and the same neural network layers, the results of PyTorch are considerably higher compared to TenserFlow, which can be seen from the lambda parameters of the equations (Fig. 7, 8).

D. The Complex Ginzburg-Landau Equation-Based PINN

The PINN, which incorporates the complex Ginsburg-Landau equation as its loss function, is realized through the utilization of the PyTorch framework. A moderately shallow, fully connected neural network is employed to approximate the state function, capitalizing on the universal approximation theorem. The entire neural network leverages automatic differentiation for the construction of the loss function. To optimize the state loss function, the Adaptive Moment Estimation (ADAM) algorithm is utilized, while the Learning with LION, which is an evolved sign momentum optimization technique [9], is employed for the optimization of the custom loss function. Although several optimizers were evaluated, some, such as ADAM, exhibited suboptimal performance, while others, like LBFGS, resulted in memory overflow, exceeding the capacity of the computational resources. In the LION optimization process, a learning rate of 1 and a weight decay of $1e-4$ were applied; the latter is a regularization technique that helps mitigate overfitting by penalizing large weights. Our computational setup permitted $10,000$ iterations for data-driven discovery related to the complex Ginzburg-Landau equation, attempting to discern the μ parameter (the coefficient preceding A in the equation, typically 1), which signifies the linear growth rate of the system. Although the results were not entirely satisfactory, this endeavor marks the beginning of a promising avenue, and we anticipate that incorporating suitable regularization techniques will yield improved outcomes.

Correct PDE	u_t + u u_x - 0.0031831 u_{xx} = 0
Identified PDE (clean data)	u_t + 0.95253 u u_x - 0.0057226 u_xx = 0
Identified PDE (1% noise)	u_t + 0.95253 u u_x - 0.0057226 u_xx = 0

Fig. 7. Tensorflow: Initial and Predicted Values of λ.

Correct PDE	u_t + u u_x - 0.0031831 u_{xx} = 0
Identified PDE (clean data)	u_t + 0.99898 u u_x - 0.0031542 u_xx = 0
Identified PDE (1% noise)	u_t + 0.99898 u u_x - 0.0031542 u_xx = 0

Fig. 8. PyTorch: Initial and Predicted Values of λ.

The main equation of Ginzburg-Landau equation is given by:

$$\partial_t A = A + (1 + ib)\Delta A - (1 + ic)|A|^2 A$$

where A is a complex function reliant on the (scaled) time t and spatial coordinates $\overrightarrow{x}$ (typically in lower dimensions $D = 1$ or 2), with real parameters b and c representing linear and nonlinear dispersion. This equation appears in physics, particularly as a "modulation," "envelope," or "amplitude" equation. It provides a streamlined, universal depiction of "weakly nonlinear" spatio-temporal occurrences in expansive $(\overrightarrow{x})$ continuous media with diverse linear dispersion characteristics (elaborated on later) and which stay invariant during a global gauge transformation (multiplying A by $exp(i\Phi)$). This symmetry usually arises when A embodies the (slowly changing) amplitude of a phenomenon that is periodic in a minimum of one variable (space and/or time) due to the system's translational invariance [10].

The above equation can be reduced to:

$$\partial_t A = \mu A + \Delta A - |A|^2 A$$

which is also known as the "Complex Nonlinear Diffusion Equation," making a connection to the Nonlinear Schrödinger Equation. Instances of such instability include Rayleigh-Bénard convection in both simple and complex fluids, Taylor-Couette flow, electroconvection in liquid crystals, among others. The related equation, developed by Newell and Whitehead (1969) and Segel (1969), was the inaugural amplitude equation to include spatial degrees of freedom. Nonetheless, it is strictly applicable to scenarios with almost parallel rolls, posing a considerable constraint in isotropic systems [10].

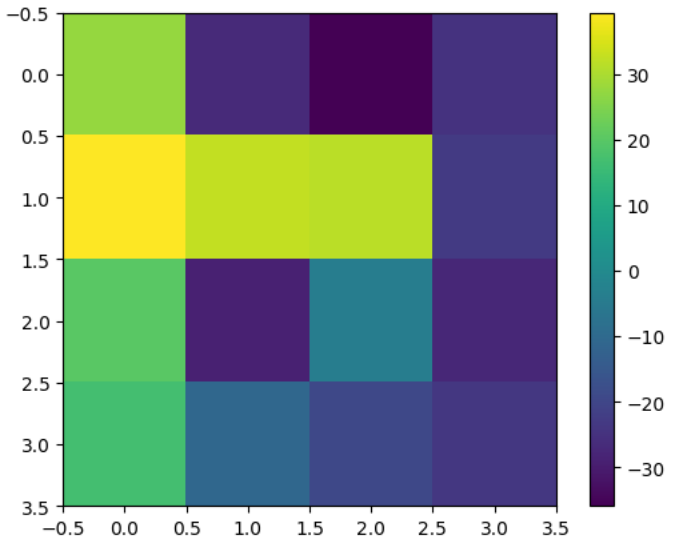

Fig. 9. Original values.

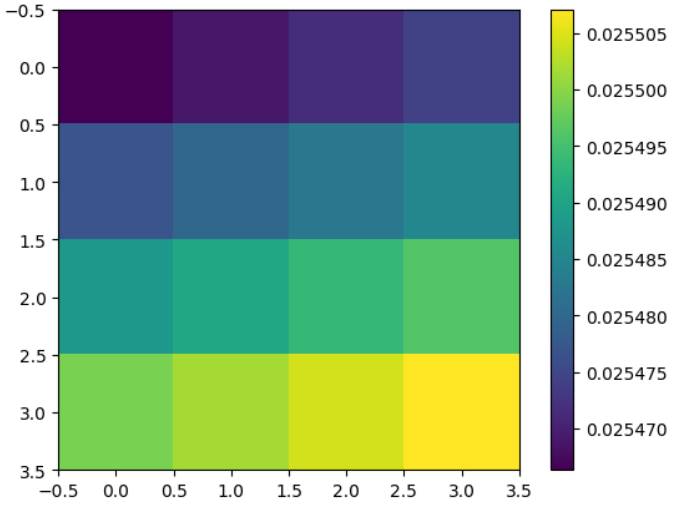

Fig. 10. Predicted values.

It is obvious from Figures 9 and 10 that the model does not work well, as it is new in the sphere. It was trained on a NVIDIA's GTX 1050 TI GPU

Figure 11 shows the losses of Ginzburg-Landau equation, and the next one is the normalized version of the losses done through the following formula:

$$X_{norm} = \frac{(X - X_{min})}{(X_{max} - X_{min})}.$$

IV. Conclusion

In conclusion, this study demonstrates the potential of Physics-Informed Neural Networks (PINNs) as a powerful tool for addressing nonlinear problems in computational science. By integrating structured prior information, PINNs are capable of efficiently learning from limited data and adhering to

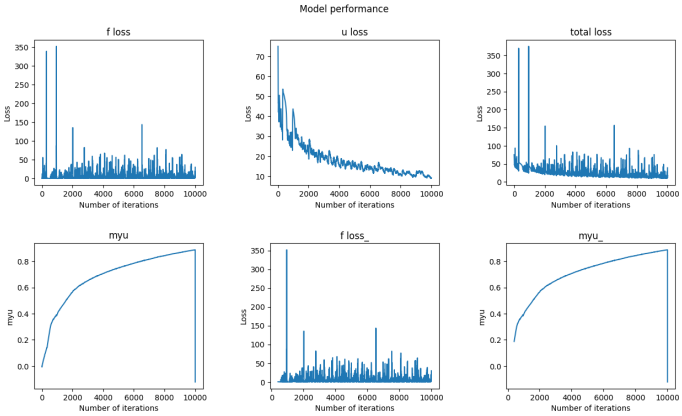

Fig. 11. Losses by Ginzburg-Landau Equation.

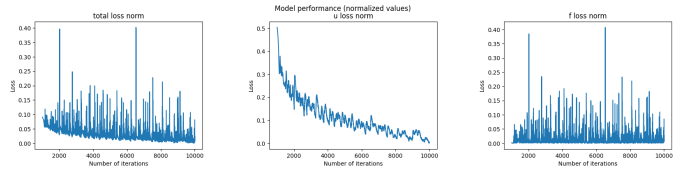

Fig. 12. Normalized Losses by Ginzburg-Landau Equation.

underlying physical laws. Comparing the implementations of PINNs in TensorFlow and PyTorch, it becomes evident that the PyTorch version offers greater adaptability, superior performance, and improved stability. While TensorFlow may be more suitable for incorporating custom features within neural network architectures, PyTorch facilitates rapid prototyping and emerges as the preferred choice for PINNs implementation. This work lays the foundation for a new paradigm in modeling and computation, combining deep learning with well-established developments in mathematical physics, opening up possibilities for novel numerical solvers, data-driven approaches, and advancements in data-efficient learning machines.

References

[1] M. Raissi, P. Perdikaris, G.E. Karniadakis, Inferring solutions of differential equations using noisy multi-fidelity data, J. Comput. Phys. 335 (2017) 736–746.

[2] M. Raissi, P. Perdikaris, G.E. Karniadakis, Machine learning of linear differential equations using Gaussian processes, J. Comput. Phys. 348 (2017) 683–693.

[3] H. Owhadi, Bayesian numerical homogenization, Multiscale Model. Simul. 13 (2015) 812–828

[4] C.E. Rasmussen, C.K. Williams, Gaussian Processes for Machine Learning, vol. 1, MIT Press, Cambridge, 2006

[5] H. Owhadi, C. Scovel, T. Sullivan, et al., Brittleness of Bayesian inference under finite information in a continuous world, Electron. J. Stat. 9 (2015) 1–79.

[6] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (1989) 359–366.

[7] A.G. Baydin, B.A. Pearlmutter, A.A. Radul, J.M. Siskind, Automatic differentiation in machine learning: a survey, 2015, arXiv:1502.05767

[8] C. Basdevant, M. Deville, P. Haldenwang, J. Lacroix, J. Ouazzani, R. Peyret, P. Orlandi, A. Patera, Spectral and finite difference solutions of the Burgers equation, Comput. Fluids 14 (1986) 23–41.

[9] Chen, X., Liang, C., Huang, D., Real, E., Wang, K., Liu, Y., Pham, H., Dong, X., Luong, T., Hsieh, C.J. and Lu, Y., Le, Q. V. (2023). Symbolic discovery of optimization algorithms. arXiv preprint arXiv:2302.06675.
[10] Aranson, I. S., Kramer, L. (2002). The world of the complex Ginzburg-Landau equation. Reviews of modern physics, 74(1), 9