International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 05 | November - 2022 Impact Factor: 7.185 ISSN: 2582-3930

A Review Paper on Physics Informed Neural Networks (PINNs)

Rohit Chavan[1], Pranay Nimse[2], Shubham Patil[3], Sharvari Dabhade[4]

Prof. Abhay Gaidhani

Department of Computer Engineering SITRC, Nashik

Abstract - Physics-Informed Neural Networks (PINNs) are neural networks (NNs) that encode model equations such as Partial Differential Equations (PDE) and physical laws as a component of the neural network. PDEs, fractional equations, integral-differential equations, and stochastic PDEs are now solved using PINNs. This novel methodology evolved as a multi-task learning framework in which a NN must fit observed data while decreasing a PDE residual. According to the study, the majority of research has focused on customizing the PINN using various activation functions, gradient optimization techniques, neural network structures, and loss function structures. Despite the wide range of applications for which PINNs have been used, advancements are still possible by demonstrating their ability to be more feasible in some contexts, most notably theoretical issues that remain unresolved.

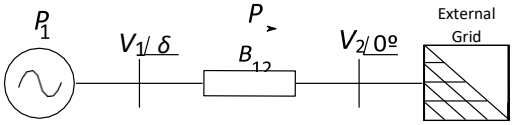
Keywords: deep learning, neural network, power system dynamics, power flow, system inertia

INTRODUCTION

Machine learning techniques demonstrate impressive results for a range of highly complex tasks, especially where an accurate mathematical representation of the problem cannot be obtained. Applications include image recognition, robotics, weather forecasting, and others [1]. In power systems, decision trees and neural networks have been shown to solve computational problems both in dynamics and optimization at a fraction of the time required by traditional approaches, being up to three order of magnitude faster.

Up to this point, however, machine learning methods applied to power systems (and other physical systems) were largely agnostic to the underlying physical model. This made them heavily dependent on the quality of the training data, it required large training datasets, and oftentimes complex neural network structures. Despite recent efforts for efficient creation of datasets with encouraging results [7], [8], generating the required training dataset size still requires substantial computational effort. In this work, inspired by [9], [10], we reduce the dependency on training data and complex neural network structures by exploiting *inside the neural network training* the underlying physical laws described by power system models.

This is the first work, to our knowledge, that proposes physics-informed neural networks for power system applications. It introduces a neural network training framework that can exploit the underlying physical laws and the available power system models both for steady-state and dynamics. Following recent approaches reported in [9], [10], we incorporate the power system differential and algebraic equations.



METHODOLOGY

A. Physical Model for Power System Dynamics

B. Power system dynamics, in their simplest and most common form, are described by the swing equation, neglecting transmission losses and bus voltage deviations. For each generator k, the resulting system of equations can then be represented.

C. Single Machine Infinite Bus (SMIB) System: The single machine infinite bus system, shown in Fig. 1, has been widely used to understand and analyze the fundamental dynamic phenomena occurring in power systems. As the focus of this paper is on the introduction of physics-informed neural networks for power systems, we will use this system as a guiding example. Note though that our proposed framework is general. Future work will focus on larger, more complex systems. The swing equation (1) for the SMIB system is given by:

$$m_1\ddot{\delta} + d_1\dot{\delta} + B_{12}V_1V_2\sin(\delta) - P_1 = 0$$

In the rest of this paper, we will show how physics-informed neural networks can accurately estimate both rotor angle δ and frequency $\dot{\delta}$ while P_1 varies within $[P_{\min}, P_{\max}]$, and can identify uncertain parameters such as m_1 and d_1.

D. Physics-Informed Neural Networks

E. In the following, we explain the general architecture of physics-informed neural networks, and detail its application to the SMIB system. Feed-forward neural networks are composed of the input layer, fully connected hidden layers having a non-linear activation function at each neuron, and the output layer. Between each layer a weight matrix **W** and bias **b** is applied.

F. General structure of a physics-informed neural network: it predicts the output $u(t, x)$ given inputs x and t. Then, using automatic differentiation [11] of the same neural network, the partial derivatives of $u(t, x)$ are computed, and $f(t, x)$ is evaluated. The parameters λ are either assumed to be known, or are optimized as part of the neural network training. During training, the neural network weights and biases are adjusted according to loss function (5), which

minimizes the deviation of both output prediction $u(t, x)$ from ground truth and $f(t, x)$ from 0.

The work in [10] introduced a framework for physics-informed neural networks which we will rely on in the following. Considering physical laws during training allows to bound the space of admissible solutions to the neural network parameters, which translates to a lower requirement in both the amount of training data and neural network size.

Following notation similar to Ref. [10], the general form of the functions that the physics-informed neural network can approximate is;

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

where $u(t, x)$ is the solution and $N[u; \lambda]$ is a nonlinear operator connecting the state variables u with the system parameters λ. The term t denotes time and x the system input. The domain Ω can be bounded based on prior knowledge of the dynamical system and $[0, T]$ is the time interval within which the system evolves. The model parameters λ can be constant or unknown. In case λ is unknown, the problem of approximating function (3) becomes a problem of system identification, where we seek parameters λ for which the expression in (3) is satisfied. To enforce the physical law describing the dynamical system we define the physics-informed neural network $f(t, x)$.

Note that if the system parameters λ are known the nonlinear operator $N[u, \lambda]$ simplifies to $N[u]$. A neural network is used to predict $u(t, x)$ based on the inputs t and x. To determine $f(t, x)$, we use automatic differentiation [11] of the components of the neural network predicting $u(t, x)$. Based on this, we compute the required derivatives of $u(t, x)$ with respect to time t and system inputs x. As a result, the neural network predicting$f(t, x)$ has the same parameters compared to the neural network predicting $u(t, x)$, but different activation functions. The shared parameters of the two neural networks are optimized by minimizing the loss function.
Number of collocation points and training data influence the prediction accuracy and the computational time to optimize the loss function. The error MSE_u enforces the boundary conditions of the independent variables x and MSE_f enforces the physics of the dynamical system imposed by the condition (3), i.e. it penalizes deviations of the predicted physical law. Given a training data set and known system parameters λ, we seek to find the parameters (weights and biases) of the neural networks which minimize (5). If the parameters λ are unknown, we train for the same objective but consider the system parameters as additional variables.

Physics-informed neural networks capturing power system dynamics:

We show how physics-informed neural networks can be used to chanical power P_1. We assume that the system parameters $\lambda: = m_1, d_1, B_{12}$ are known and the voltages V_1 and V_2 are fixed. As a result, the system input is defined as $x: = P_1$. In contrast with conventional numerical solvers, which requirethe conversion of higher-order ordinary differential equations (ODEs) to first-order in order to solve them (by introducing additional variables), physics-informed neural networks can directly incorporate higher-order ODEs, as we show in (7).
Incorporating (2) to the neural network, function (4) is given

$$u(t, x) := \delta(t, P_1),$$

$$f_\delta(t, P_1) = m_1\ddot{\delta} + d_1\dot{\delta} + B_{12}V_1V_2\sin(\delta) - P_1$$

Data-driven discovery of inertia and damping coefficients:

Information about power system parameters such as system inertia is of significant importance for system operatorsto prevent large frequency deviations and maintain frequency stability. As described in [16], due to varying generation of converter-connected renewable energy sources, the inertia level of power systems becomes uncertain and has to be estimated (or predicted) at regular time intervals [17]. Physics-informed neural networks can be used to address the problem of system identification and data-driven discovery of partial differential equations. For this case, we define m_1 and d_1 as unknown parameters in (7). The structure of the physics-informed neural network remains the same, with the only difference that a subset of the system parameters λ are now treated as additional variables when minimizing (5) during neural network training.

SIMULATION

Simulation Setup

Besides an initial training set, to assess the neural network performance we also need an extensive test data set. To create the training and test data sets we use the numerical solver *ode*45 in MATLAB with a time step of 0.1*s* and time interval $T = [0, 20s]$, resulting in 201 time steps for each trajectory. The voltage magnitudes V_1 and V_2 are equal to 1 p.u. and $B_{12} = 0.2$ p.u. In our first case study, we assume system inertia and damping are known, and that the system is not at an equilibrium. Assuming an uncertain active power input in the range $P_1 = [0.08, 0.18]$ and initial values for δ and ω equal to 0.1 rad and 0.1 rad/s, we generate 100 trajectories. As a result, our entire test and training dataset consists of 20′100 samples.

In our second case study, inertia and damping are also unknown parameters. Given scattered observed data about active power, frequency and angle measurements, our goal is to identify the parameters m_1 and d_1 of (7), as well as to obtain the trajectory of δ. Considering that the levels of inertia and damping vary, we assign 10 different values to m_1 and d_1 that lie within the range of [0.1, 0.4] and [0.05, 0.15], respectively. To this end, for each of the 10 pairs m_1, d_1 we generate 40 trajectories.

A. Data-driven solution of frequency dynamics through physics-informed neural networks

B. The following parameters were selected to obtain the lowest L_2 error on the test data: we select a set of N_u = 40 randomly distributed initial and boundary data across the entire spatiotemporal domain, N_f = 8′000 collocation points, and a 5-layer neural network with 10 neurons per hidden layer. Observe that compared to conventional neural network approaches, we onlyneed a very small amount of samples (N_u = 40). Increasing N_u in our simulations, led to over-fitting to the training data. Training took 223 seconds and the relative L_2 error between exact and predicted solutions on the 11′600-points test dataset is $1.34\ 10^{-2}$. Fig. 3 depicts the comparison between the predicted and the actual trajectory of the angle $\delta(t)$ and the frequency $\omega(t)$. The best

and worst δ, ω estimation during different active power inputs P_1 in terms of L_2 error on both.

C. Predicting both angle δ and frequency ω as separate neural network outputs:

Within our investigations, we also attempted to train a physics-informed neural network that considers δ and ω as separate outputs, essentially setting $\boldsymbol{f}_\omega = \dot{\delta}$, ω and $\boldsymbol{f}_\delta = m_1\dot{\omega} + d_1\omega + \boldsymbol{B}_{12}V_1V_2\sin(\delta)$ P_1.

To obtain the lowest L_2 error in this case, we had to select again a set of $N_u = 40$ randomly distributed initial and boundary data. Considering that $\delta(t)$ and $\omega(t)$ are predicted as separate outputs, the relative L_2 errors between the exact and predicted solutions are 9.43 10^{-2} and 1.51 10^{-1}, respectively, and are higher than for the NN_δ structure. It becomes obvious that the neural network architecture with the single output δ (and subsequent numerical differentiation to determine ω) is preferable in terms of training time and predictive accuracy.

At the first occurrence of an acronym, spell it out followed by the acronym in parentheses, e.g., charge-coupled diode (CCD).

D. Data-driven discovery of inertia and damping coefficients through physics-informed neural networks

E. In this subsection, we evaluate the performance of the physics-informed neural network to predict system inertia and damping from observed trajectories. In this case study, we assume that m_1 and d_1 are unknown, and instead we have a set of limited training data points t, P_1, δ. Contrary to the usual practice of first training a neural network and then using it, our objective here is exploit the physics-informed neural network training procedure to determine m_1 and d_1. To illustrate the effectiveness of this approach, we perform this analysis for 10 different pairs of m_1, d_1 and evaluate the average predictive accuracy. We select a set of N_u = 100 randomly distributed points across the spatiotemporal domain from the exact solutions of (2) for each inertia level. A 5-layer neural network with 30 neurons per hidden layer is trained for each inertia level with the corresponding trajectories in order to predict the system parameters and $\delta(t)$. The resulting average errors for predicting m_1 and d_1 over the 10 different cases are 0.74% and 1.28%, respectively. The average training time of the neural network to identify the system parameters was less than 60 seconds. This means that with a limited training dataset, and within 60 seconds, we can accurately predict the inertia and damping level of a system. Considering that the swing equation (2) is often used to approximate the aggregate dynamic behavior of large power systems, these results demonstrate that physics-informed neural network show substantial potential to not only accurately derive δ and ω but also predict both system inertia and damping. Last but not least, the relative L_2 errors between the exact and predicted solutions for the phase angle are less than 10^{-1} over the 10 different cases of m_1, d_1. This shows the potential of physics informed neural networks to be used as a dynamic state estimator, when the model parameters are unknown.

DISCUSSION AND OUTLOOK

This work introduces for the first time in power systems a neural network training procedure that explicitly considers the underlying differential and algebraic equations describing power system behavior. This unlocks a series of opportunities in power systems, as physics-informed neural networks may be able to accurately determine the solution of differential algebraic sets of equations several orders of magnitude faster than traditional methods relying on numerical integration. Still,to unlock this potential, there are several challenges to be addressed.

Number of training data: Besides the limited number of training data, physics-informed neural networks as described in this paper need to generate a substantial number of collocation points. In our case studies, we used $N_u = 40$ points as input data and $N_f = 8'000$ collocation points. It is expected that for larger systems, .a much larger numberof collocation points will be necessary, which will resultto a longer training time. In our future work, we plan to investigate methods using Runge-Kutte integration schemes such as the ones proposed in [10] which can eliminate the need for collocation points.

Scalability: Although the swing equation is a good first approximation for first-swing instability, and single- machine infinite-bus systems are still used as aggregate modelsof large power systems, we still need to explore what are the computational needs if we were to apply these methods in large scale power systems and how to address the associated challenges related to the neural network training. Particularly, the comparison with numerical solvers and approximation techniques like polynomial fits will serve as a benchmark.

Range of applications: As shown in this paper, physics-informed neural networks can determine two orders of magnitude faster the rotor angle and frequency at any time instant for uncertain power inputs. At the same time, they can accurately identify uncertain parameters such as inertia and damping. Future applications must also assess cases that include both stable and unstable equilibria, a wide range of different dynamic phenomena, including small-signal stability, voltage stability and converter dynamics [18], discrete events, such as protection actions, as well as power system optimization, among numerous others. In our simulation study, we observed high accuracy for a single stable swing prediction, but for different regimes such as multiple oscillations or unstable conditions, different physics-informed neural networks might have to be trained. We also need to examine if such neural networks can capture discrete events, such as protection actions, or if we need to develop a hybrid approach, using physics-informed neural networks as a numerical solver only during the continuous dynamics before and after a discrete event. For power system applications, physics-informed neural networks can (and should) be combined with neural network verification methods, see [19]. In this way, they would no longer be considered a black box, but instead we would be able to extract formal guarantees for their behavior.

Understanding deep learning: In addition to enhancing the trainability and generalization of ML models, physical principles are also being used to provide theoretical insight and elucidate the inner mechanisms behind the surprising effectiveness of deep learning. For example, in REFS the authors use the jamming transition of granular media to understand the double-descent phenomenon of deep learning in the over-parameterized regime. Shallow NNs can also be viewed as interacting particle systems and hence can be analyzed in the probability measure space with mean-field theory, instead of the high-dimensional parameter space. Another work rigorously constructed an exact mapping from the variational renormalization group to deep learning architectures based on restricted Boltzmann machines. Inspired by the successful density matrix renormalization group algorithm developed in physics, REF proposed a framework for applying quantum-inspired tensor networks to multi-class supervised learning tasks, which introduces consider- able savings in computational cost. Reference studied the landscape of deep networks from a statistical physics viewpoint, establishing an intuitive connection between NNs and the spin-glass models. In parallel, information propagation in wide DNNs has been studied based on dynamical systems theory, providing an analysis of how network initialization determines the propagation of an input signal through the network, hence identifying a set of hyper-parameters and activation functions known as the 'edge of chaos' that ensure information propagation in deep networks.

APPLICATIONS

In this section, we discuss some of the capabilities of physics-informed learning through diverse applications. Our emphasis is on inverse and ill posed problems, which are either difficult or impossible to solve with conventional approaches. We also present several ongoing efforts on developing open-source software for scientific ML.

Observational biases

Observational data are perhaps the foundation of the recent success of ML. They are also conceptually the simplest mode of introducing biases in ML. Given sufficient data to cover the input domain of a learning task, ML methods have demonstrated remarkable power in achieving accurate interpolation between the dots, even for high-dimensional tasks. For physical systems in particular, thanks to the rapid development of sensor networks, it is now possible to exploit a wealth of variable fidelity observations and monitor the evolution of complex phenomena across several spatial and temporal scales. These observational data ought to reflect the underlying physical principles that dictate their generation, and, in principle, can be used as a weak mechanism for embedding these principles into an ML model during its training phase. Examples include NNs proposed in REFs. However, especially for over-parameterized deep learning models, a large volume of data is typically necessary to reinforce these biases and generate predictions that respect certain symmetries and conservation laws. In this case, an immediate difficulty relates to the cost of data acquisition, which for many applications in the physical and engineering sciences could be prohibitively large, as observational data may be generated via expensive experiments or large-scale computational models.

Inductive biases

Another school of thought pertains to efforts focused on designing specialized NN architectures that implicitly embed any prior knowledge and inductive biases associated with a given predictive task. Without a doubt, the most celebrated example in this category are convolutional NNs which have revolutionized the field of computer vision by craftily respecting invariance along the groups of symmetries and distributed pattern representations found in natural images Additional representative examples include graph neural networks (GNNs) equivariant networks, kernel methods such as Gaussian processes and more general PINs with kernels that are directly induced by the physical principles that govern a given task. Convolutional net- works can be generalized to respect more symmetry groups, including rotations, reflections and more general gauge symmetry transformations. This enables the development of a very general class of NN architectures on manifolds that depend only on the intrinsic geometry, leading to very effective models for computer vision tasks involving medical images, climate pattern segmentation and others. Translation-invariant representations can also be constructed via wavelet-based scattering transforms, which are stable to deformations and preserve high-frequency information Another in many-body systems. A similar example is the equivariant transformer networks, a family of differentiable mappings that improve the robustness of models for predefined continuous transformation.

Hybrid approaches

The aforementioned principles of physics-informed ML have their own advantages and limitations. Hence, it would be ideal to use these different principles together, and indeed different hybrid approaches have been proposed. For example, non-dimensionalization can recover characteristic properties of a system, and thus it is beneficial to introduce physics bias via appropriate non-dimensional parameters, such as Reynolds, Froude or Mach numbers. Several methods have been proposed to learn operators that describe physical phenomena for example, DeepONets have been demonstrated as a powerful tool to learn nonlinear operators in a supervised data-driven manner. What is more exciting is that by combining DeepONets with physics encoded by PINNs, it is possible to accomplish real-time accurate predictions with extrapolation in multiphysics applications such as electro-convection and hypersonic. However, when a low-fidelity model is available, a multi-fidelity strategy can be developed to facilitate the learning of a complex system. For example, REF. combines observational and learning biases through the use of large-eddy simulation data and con- strained NN training methods to construct closures for lower-fidelity Reynolds-averaged Navier–Stokes models of turbulent fluid flow. Additional representative use-cases include the multi-fidelity NN used in REF. to extract material properties from instrumented indentation data, the PINs in REF. used to discover constitutive laws of non-Newtonian fluids from rheological data, and the

coarse-graining strategies proposed in REF. Even if it is not possible to encode the low-fidelity model into the learning directly, the low-fidelity model can be used through data augmentation — that is, generating a large amount of low-fidelity data via inexpensive low-fidelity models, which could be simplified mathematical models or existing computer codes, such as REF. Other representative examples include FermiNetsand graph neural operator methods. It is also possible to enforce the physics to an NN by embedding a network into a traditional numerical method (such as finite element). This approach was applied to solve problems in many different fields, including nonlinear dynamical systems computational mechanics to model constitutive relations, subsurface mechanics stochastic inversion and more.

Connections to kernel methods

Many of the presented NN-based techniques have a close asymptotic connection to kernel methods, which can be exploited to produce new insight and understanding. For example, as demonstrated in REFs, the training dynamics of PINNs can be understood as a kernel regression method as the width of the network goes to infinity. More generally, NN methods can be rigorously interpreted as kernel methods in which the underlying warping kernel is also learned from data. Warping kernels are a special kind of kernels that were initially introduced to model non-stationary spatial structures in geostatistics and have been also used to interpret residual NN models. Furthermore, PINNs can be viewed as solving PDEs in a reproducing kernel Hilbert space spanned by a feature map (parametrized by the initial layers of the network), where the latter is also learned from data. Further connections can be made by studying the intimate connection between statistical inference techniques and numerical approximation. Existing works have explored these connections in the context of solving PDEs and inverse problems optimal recovery and Bayesian numerical analysis Connections between kernel methods and NNs can be established even for large and complicated architectures, such as attention-based transformers, whereas operator-valued kernel methods could offer a viable path of analyzing and interpreting deep learning tools for learning nonlinear operators. In summary, analyzing NN models through the lens of kernel methods could have considerable benefits, as kernel methods are often interpretable and have strong theoretical foundations, which can subsequently help us to understand when and why deep learning methods may fail or succeed.

Connections to classical numerical methods

Classical numerical algorithms, such as Runge–Kutta methods and finite-element methods, have been the main workhorses for studying and simulating physical systems in silico. Interestingly, many modern deep learning models can be viewed and analyzed by observing an obvious correspondence and specific connections to many of these classical algorithms. In particular, several architectures that have had tremendous success in practice are analogous to established strategies in numerical analysis. Convolutional NNs, for example, are analogous to finite different stencils in translation- ally equivariant PDE discretization and share the same structures as the multigrid method; residual NNs (ResNets, networks with skip connections) are analogous to the basic forward Euler discretization of linear finite-element method Such analogies can provide insights and guidance for cross-fertilization, and pave the way for new 'mathematics-informed' meta-learning architectures. For example, REF. pro- posed a discrete-time NN method for solving PDEs that is inspired by an implicit Runge–Kutta integrator: using up to 500 latent stages, this NN method can allow very large time-steps and lead to solutions of high accuracy.

Tackling high dimensionality

Deep learning has been very successful in solving high-dimensional problems, such as image classification with fine resolution, language modelling, and high-dimensional PDEs. One reason for this success is that DNNs can break the curse of dimensionality under the condition that the target function is a hierarchical composition of local functions. For example, in REF. the authors reformulated general high-dimensional parabolic PDEs using backward stochastic differential equations, approximating the gradient of the solution with DNNs, and then designing the loss based on the discretized stochastic integral and the given terminal condition. In practice, this approach was used to solve high-dimensional Black–Scholes, Hamilton–Jacobi–Bellman and Allen–Cahn equations GANs have also proven to be fairly successful in generating samples from high-dimensional distributions in tasks such as image or text generation. As for their application to physical problems, in REF the authors used GANs to quantify parametric uncertainty in high-dimensional stochastic differential equations, and in REF. GANs were used to learn parameters in high-dimensional stochastic dynamics. These examples show the capability of GANs in modelling high-dimensional probability distributions in physical problems. Finally, in REFs it was demonstrated that even for operator regression and applications to PDEs, deep operator networks (DeepONets) can tackle the curse of dimensionality associated with the input space. The third source of uncertainty refers to the limitation of the learning models— for example, the approximation, training and generalization errors of NNs — and is usually hard to rigorously quantify. In REF. a convolutional encoder–decoder NN is used to map the source term and the domain geometry of a PDE to the solution as well as the uncertainty, trained by a probabilistic supervised learning procedure with training data coming from finite-element methods. Notably, a first attempt to quantify the combined uncertainty from learning was given in REF. using the dropout method of REF. and, due to physical random- ness, using arbitrary polynomial chaos. An extension to time-dependent systems and long-time integration was reported in REF.[42]: it tackled the parametric uncertainty using dynamic and bi-orthogonal modal decomposition of the stochastic PDE, which are effective methods for long-term integration of stochastic systems.

Application to geophysics

Physics-informed learning has also been applied to various geophysical inverse problems. The work in REF, estimates subsurface properties, such as rock permeability and porosity, from seismic data by coupling NNs with full-waveform inversion, subsurface flow processes and rock physics models. Furthermore, in REF, it was demonstrated that by combining DNNs and numerical PDE solvers as we discussed in the section on hybrid approaches, physics-informed learning is capable of solving a wide class of seismic inversion problems, such as velocity estimation, fault rupture imaging, earthquake location and source–time function retrieval.

Software

To implement PINNs efficiently, it is advantageous to build new algorithms based on the current ML libraries, such as TensorFlow, PyTorch, Keras and JAX Several software libraries specifically designed for physics-informed ML have been developed and are contributing to the rapid development of the field (TABLE 1). At the present time, some of the actively developed libraries include DeepXDE SimNet, PyDEns NeuroDiffEqNeuralPDE SciANN and ADCME. Because Python is the dominant programming language for ML, it is more convenient to use Python for physics-informed ML, and thus most of these libraries are written in Python, except the NeuralPDE and ADCME, which are written in Julia. All these libraries use the automatic differentiation mechanism provided in other software such as TensorFlow. Some of these libraries (such as DeepXDE and SimNet) can be used as a solver, that is, users only need to define the problem and then the solver will deal DeepXDE not only solves integer-order ODEs and PDEs, but it can also solve integro-differential equations and fractional PDEs. DeepXDE supports complex domain geometries via the technique of constructive solid geometry, and enables the user code to stay com- pact, resembling closely the mathematical formulation. DeepXDE is also well-structured and highly configurable, since all its components are loosely coupled.

Which model, framework, algorithm to use?

With a growing collection of methodologies and software tools, a series of questions naturally arises: given a physical system and/or governing law and some observational data, which ML framework should one use? Which training algorithm to choose? How many training samples to consider? Although at present there are no rule-of-thumb strategies for answering these questions, and some degree of experience is required to set up a physics-informed ML model properly, meta-learning. The choices intimately depend on the specific task that needs to be tackled. In terms of providing a high-level taxonomy, we note that PINNs are typically used to infer a deterministic function that is compatible with an underlying physical law when a limited number of observations is available (either initial/boundary conditions or other measurements). The underlying architecture of a PINNs model is determined by the nature of a given problem: multi-layer perceptron architectures are generally applicable but do not encode any specialized inductive biases, convolutional NN architectures are suitable for gridded 2D domains, Fourier feature networks are suitable for PDEs whose solution exhibits high frequencies or periodic boundaries, and recur- rent architectures are suitable for non-Markovian and time-discrete problems.

CONCLUSIONS

We've seen how machine learning provides a new way of conducting scientific research by emphasizing data-driven learning. By incorporating existing physical principles into machine learning, we can create more powerful models that learn from data and build on our existing scientific knowledge. We have introduced physics-informed neural networks, a new class of universal function approximators that is capable of encoding any underlying physical laws that govern a given data-set, and can be described by partial differential equations. In this work, we design data-driven algorithms for inferring solutions to general nonline ar partial differential equations, and constructing computationally efficient physics-informed surrogate models. The resulting methods showcase a series of promising results for a diverse collection of problems in computational science,and open the path for endowing deep learning with the powerful capacity of mathematical physics to model the world around us. As deep learning technology is continuing to grow rapidly both in terms of methodological and algorithmic developments, we believe that this is a timely contribution that can benefit practitioners across a wide range of scientific domains. Specific applications that can r eadily enjoy these benefits include, but are not limited to, data-driven forecasting of physical processes, model predictive control, multi-physics/multi-scale modelling and simulation.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521,no. 7553, pp. 436–444, 2015.

[2] L. A. Wehenkel, *Automatic learning techniques in power systems*. Springer Science & Business Media, 2012.

[3] B. Donnot, I. Guyon, M. Schoenauer, P. Panciatici, and A. Marot, “Introducing machine learning for power system operation support,” *CoRR, arXiv preprint arXiv:1709.09527*, 2017.

[4] M. Sun, I. Konstantelos, and G. Strbac, “A deep learning-based feature extraction framework for system security assessment,” *IEEE Transac- tions on Smart Grid*, vol. 10, no. 5, pp. 5007–5020, Sep. 2019.

[5] J. H. Arteaga, F. Hancharou, F. Thams, and S. Chatzivasileiadis, “Deep learning for power system security assessment,” in *2019 IEEE Milan PowerTech*, June 2019, pp. 1–6.

[6] F. Fioretto, T. W. K. Mak, and P. V. Hentenryck, “Predicting ac optimal power flows: Combining deep learning and lagrangian dual methods,” *arXiv preprint arXiv:1909.10461*, 2019.

[7] F. Thams, A. Venzke, R. Eriksson, and S. Chatzivasileiadis, “Efficient database generation for data-driven security assessment of power sys- tems,” *IEEE Transactions on Power Systems*, pp. 1–1, 2019.

[8] A. Venzke, D. K. Molzahn, and S. Chatzivasileiadis, “Efficient creation of datasets for data-driven power system applications,” *arXiv preprint arXiv:1910.01794*,

2019.

[9] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Advances in neural information processing systems*, 2018, pp. 6571–6583.

[10] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686 – 707, 2019.

[11] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind,"Automatic differentiation in machine learning: a survey," *Journal of machine learning research*, vol. 18, no. 153, 2018.

[12] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on hetero- geneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[13] J. Zhao *et al.*, "Power system dynamic state estimation: Motivations, definitions, methodologies, and future work," *IEEE Trans. on Power Syst.*, vol. 34, no. 4, July 2019.

[14] G. S. Misyris, A. Venzke, and S. Chatzivasileiadis, "Online Appendix: Physics-Informed Neural Networks for Power Systems," 2019. [Online]. Available: https://github.com/gmisy/Physics-Informed-Neural-Networks-for-Power-Systems/

[15] S. Chatzivasileiadis, T. L. Vu, and K. Turitsyn, "Remedial actions to enhance stability of low-inertia systems," in *IEEE Power and Energy Society General Meeting 2016, Boston, MA, USA*, July 2016, pp. 1 –5.