

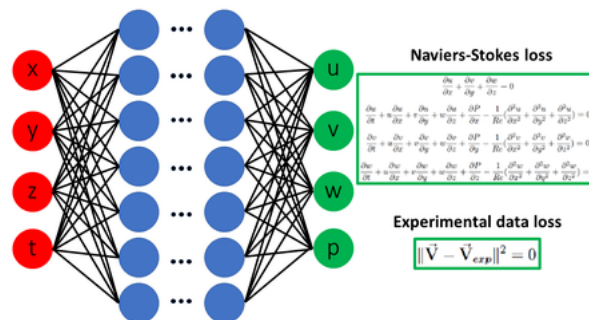
WIKIPEDIA

The Free Encyclopedia

Physics-informed neural networks

Physics-informed neural networks

Physics-informed neural networks (PINNs) are a type of universal function approximators that can embed the knowledge of any physical laws that govern a given dataset in the learning process, and can be described by partial differential equations (PDEs).^[1] They overcome the low data availability of some biological and engineering systems that makes most state-of-the-art machine learning techniques lack robustness, rendering them ineffective in these scenarios.^[1] The prior knowledge of general physical laws acts in the training of neural networks (NNs) as a regularization agent that limits the space of admissible solutions, increasing the correctness of the function approximation. This way, embedding this prior information into a neural network results in enhancing the information content of the available data, facilitating the learning algorithm to capture the right solution and to generalize well even with a low amount of training examples.



Physics-informed neural networks for solving Navier–Stokes equations

Function approximation

Most of the physical laws that govern the dynamics of a system can be described by partial differential equations. For example, the Navier–Stokes equations^[2] are a set of partial differential equations derived from the conservation laws (i.e., conservation of mass, momentum, and energy) that govern fluid mechanics. The solution of the Navier–Stokes equations with appropriate initial and boundary conditions allows the quantification of flow dynamics in a precisely defined geometry. However, these equations cannot be solved exactly and therefore numerical methods must be used (such as finite differences, finite elements and finite volumes). In this setting, these governing equations must be solved while accounting for prior assumptions, linearization, and adequate time and space discretization.

Recently, solving the governing partial differential equations of physical phenomena using deep learning has emerged as a new field of scientific machine learning (SciML), leveraging the universal approximation^[3] and high expressivity of neural networks. In general, deep neural networks could approximate any high-dimensional function given that sufficient training data are supplied.^[4] However, such networks do not consider the physical characteristics underlying the problem, and the level of approximation accuracy provided by them is still heavily dependent on careful specifications of the problem geometry as well as the initial and boundary conditions. Without this preliminary information, the solution is not unique and may lose physical correctness. On the other hand, physics-informed neural networks (PINNs) leverage governing physical equations in neural network training. Namely, PINNs are designed to be trained to satisfy the given training data as well as the imposed governing equations. In this fashion, a neural network can be guided with training data that do not necessarily need to be large and complete.^[4] Potentially, an accurate solution of partial differential equations can be found without knowing the boundary conditions.^[5] Therefore, with some knowledge about the physical characteristics of the problem and some form of training data (even sparse and

incomplete), PINN may be used for finding an optimal solution with high fidelity.

PINNs allow for addressing a wide range of problems in computational science and represent a pioneering technology leading to the development of new classes of numerical solvers for PDEs. PINNs can be thought of as a meshfree alternative to traditional approaches (e.g., CFD for fluid dynamics), and new data-driven approaches for model inversion and system identification.^[6] Notably, the trained PINN network can be used for predicting the values on simulation grids of different resolutions without the need to be retrained.^[7] In addition, they allow for exploiting automatic differentiation (AD)^[8] to compute the required derivatives in the partial differential equations, a new class of differentiation techniques widely used to derive neural networks assessed to be superior to numerical or symbolic differentiation.

Modeling and computation

A general nonlinear partial differential equations can be:

$$\mathbf{u}_t + N[\mathbf{u}; \boldsymbol{\lambda}] = 0, \quad \mathbf{x} \in \Omega, \quad t \in [0, T]$$

where $\mathbf{u}(t, \mathbf{x})$ denotes the solution, $N[\cdot; \boldsymbol{\lambda}]$ is a nonlinear operator parametrized by $\boldsymbol{\lambda}$, and Ω is a subset of \mathbb{R}^D . This general form of governing equations summarizes a wide range of problems in mathematical physics, such as conservative laws, diffusion process, advection-diffusion systems, and kinetic equations. Given noisy measurements of a generic dynamic system described by the equation above, PINNs can be designed to solve two classes of problems:

- data-driven solution
- data-driven discovery

of partial differential equations.

Data-driven solution of partial differential equations

The *data-driven solution of PDE*^[1] computes the hidden state $\mathbf{u}(t, \mathbf{x})$ of the system given boundary data and/or measurements \mathbf{z} , and fixed model parameters $\boldsymbol{\lambda}$. We solve:

$$\mathbf{u}_t + N[\mathbf{u}] = 0, \quad \mathbf{x} \in \Omega, \quad t \in [0, T].$$

By defining the residual $\mathbf{f}(t, \mathbf{x})$ as

$$\mathbf{f} := \mathbf{u}_t + N[\mathbf{u}] = 0,$$

and approximating $\mathbf{u}(t, \mathbf{x})$ by a deep neural network. This network can be differentiated using automatic differentiation. The parameters of $\mathbf{u}(t, \mathbf{x})$ and $\mathbf{f}(t, \mathbf{x})$ can be then learned by minimizing the following loss function L_{tot} :

$$L_{tot} = L_u + L_f.$$

Where $L_u = \|\mathbf{u} - \mathbf{z}\|_{\Gamma}$ is the error between the PINN $\mathbf{u}(t, \mathbf{x})$ and the set of boundary conditions and measured data on the set of points Γ where the boundary conditions and data are defined, and $L_f = \|\mathbf{f}\|_{\Gamma}$ is the mean-squared error of the residual function. This second term encourages the PINN to learn the structural information expressed by the partial

differential equation during the training process.

This approach has been used to yield computationally efficient physics-informed surrogate models with applications in the forecasting of physical processes, modeling predictive control, multi-physics and multi-scale modeling, and simulation.^[9]

Data-driven discovery of partial differential equations

Given noisy and incomplete measurements \mathbf{z} of the state of the system, the *data-driven discovery of PDE*^[6] results in computing the unknown state $\mathbf{u}(\mathbf{t}, \mathbf{x})$ and learning model parameters λ that best describe the observed data and it reads as follows:

$$\mathbf{u}_t + N[\mathbf{u}; \lambda] = 0, \quad \mathbf{x} \in \Omega, \quad t \in [0, T].$$

By defining $\mathbf{f}(\mathbf{t}, \mathbf{x})$ as

$$\mathbf{f} := \mathbf{u}_t + N[\mathbf{u}; \lambda] = 0,$$

and approximating $\mathbf{u}(\mathbf{t}, \mathbf{x})$ by a deep neural network, $\mathbf{f}(\mathbf{t}, \mathbf{x})$ results in a PINN. This network can be derived using automatic differentiation. The parameters of $\mathbf{u}(\mathbf{t}, \mathbf{x})$ and $\mathbf{f}(\mathbf{t}, \mathbf{x})$, together with the parameter λ of the differential operator can be then learned by minimizing the following loss function L_{tot} :

$$L_{tot} = L_u + L_f.$$

Where $L_u = \|\mathbf{u} - \mathbf{z}\|_{\Gamma}$, with \mathbf{u} and \mathbf{z} state solutions and measurements at sparse location Γ , respectively and $L_f = \|\mathbf{f}\|_{\Gamma}$ residual function. This second term requires the structured information represented by the partial differential equations to be satisfied in the training process.

This strategy allows for discovering dynamic models described by nonlinear PDEs assembling computationally efficient and fully differentiable surrogate models that may find application in predictive forecasting, control, and data assimilation.^{[10][11][12]}

Physics-informed neural networks for piece-wise function approximation

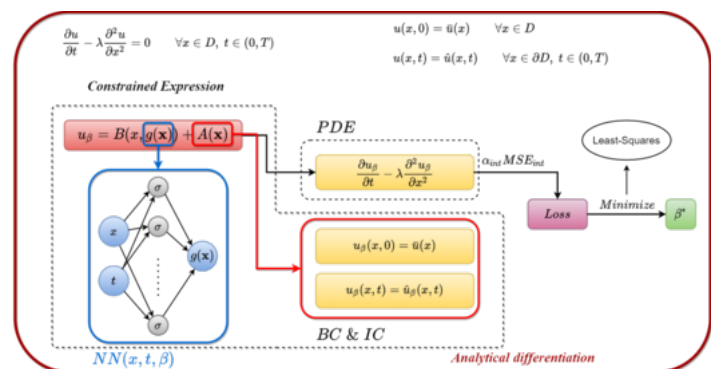
PINN is unable to approximate PDEs that have strong non-linearity or sharp gradients that commonly occur in practical fluid flow problems. Piece-wise approximation has been an old practice in the field of numerical approximation. With the capability of approximating strong non-linearity extremely light weight PINNs are used to solve PDEs in much larger discrete subdomains that increases accuracy substantially and decreases computational load as well.^{[13][14]} DPINN(Distributed physics-informed neural networks) and DPELM(Distributed physics-informed extreme learning machines) are generalizable space-time domain discretization for better approximation.^{[13][15]} DPELM is an extremely fast and lightweight approximator with competitive accuracy. Another school of thought is discretization for parallel computation to leverage usage of available computational resources.

XPINNs^[16] is a generalized space-time domain decomposition approach for the physics-informed neural networks (PINNs) to solve nonlinear partial differential equations on arbitrary

complex-geometry domains. The XPINNs further pushes the boundaries of both PINNs as well as Conservative PINNs (cPINNs),^[17] which is a spatial domain decomposition approach in the PINN framework tailored to conservation laws. Compared to PINN, the XPINN method has large representation and parallelization capacity due to the inherent property of deployment of multiple neural networks in the smaller subdomains. Unlike cPINN, XPINN can be extended to any type of PDEs. Moreover, the domain can be decomposed in any arbitrary way (in space and time), which is not possible in cPINN. Thus, XPINN offers both space and time parallelization, thereby reducing the training cost more effectively. The XPINN is particularly effective for the large-scale problems (involving large data set) as well as for the high-dimensional problems where single network based PINN is not adequate. The rigorous bounds on the errors resulting from the approximation of the nonlinear PDEs (incompressible Navier–Stokes equations) with PINNs and XPINNs are proved.^[12]

Physics-informed neural networks and functional interpolation

In the PINN framework, initial and boundary conditions are not analytically satisfied, thus they need to be included in the loss function of the network to be simultaneously learned with the differential equation (DE) unknown functions. Having competing objectives during the network's training can lead to unbalanced gradients while using gradient-based techniques, which causes PINNs to often struggle to accurately learn the underlying DE solution. This drawback is overcome by using functional interpolation techniques such as the Theory of Functional Connections (TFC)'s constrained expression, in the Deep-TFC framework, which reduces the solution search space of constrained problems to the subspace of neural network that analytically satisfies the constraints.^[18] A further improvement of PINN and functional interpolation approach is given by the Extreme Theory of Functional Connections (X-TFC) framework, where a single-layer Neural Network and the extreme learning machine training algorithm are employed.^[19] X-TFC allows to improve the accuracy and performance of regular PINNs, and its robustness and reliability are proved for stiff problems, optimal control, aerospace, and rarefied gas dynamics applications.^{[20][21][22]}



X-TFC framework scheme for PDE solution learning

This drawback is overcome by using functional interpolation techniques such as the Theory of Functional Connections (TFC)'s constrained expression, in the Deep-TFC framework, which reduces the solution search space of constrained problems to the subspace of neural network that analytically satisfies the constraints.^[18] A further improvement of PINN and functional interpolation approach is given by the Extreme Theory of Functional Connections (X-TFC) framework, where a single-layer Neural Network and the extreme learning machine training algorithm are employed.^[19] X-TFC allows to improve the accuracy and performance of regular PINNs, and its robustness and reliability are proved for stiff problems, optimal control, aerospace, and rarefied gas dynamics applications.^{[20][21][22]}

Physics-informed PointNet (PIPNet) for multiple sets of irregular geometries

Regular PINNs are only able to obtain the solution of a forward or inverse problem on a single geometry. It means that for any new geometry (computational domain), one must retrain a PINN. This limitation of regular PINNs imposes high computational costs, specifically for a wide investigation of geometric parameters in industrial designs. Physics-informed PointNet (PIPNet)^[23] is fundamentally the result of a combination of PINN's loss function with PointNet.^[24] In fact, instead of using a simple fully connected neural network, PIPNet uses PointNet as the core of its neural network. PointNet has been primarily designed for deep learning of 3D object classification and segmentation by the research group of Leonidas J.

Guibas. PointNet extracts geometric features of input computational domains in PIPN. Thus, PIPN is able to solve governing equations on a multiple computational domains (rather than only a single domain) with irregular geometries, simultaneously. The effectiveness of PIPN has been shown for incompressible flow and heat transfer.^[23]

Limitations

Translation and discontinuous behavior are hard to approximate using PINNs.^[13] They fail when solving differential equations with slight advective dominance.^[14] They also fail to solve a system of dynamical systems and hence has not been a success in solving chaotic equations.^[25] One of the reasons behind the failure of the regular PINNs is soft-constraining of Dirichlet and Neumann boundary conditions which pose multi-objective optimization problem. This requires the need for manually weighing the loss terms to be able to optimize. Another reason is getting optimization itself. Posing PDE solving as an optimization problems brings in all the problems that are faced in the world of optimization, the major one being getting stuck at a local optimum pretty often.^{[14][26]}

References

1. Raissi, Maziar; Perdikaris, Paris; Karniadakis, George Em (2017-11-28). "Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations". arXiv:1711.10561 (<https://arxiv.org/abs/1711.10561>) [cs.AI (<https://arxiv.org/archive/cs/AI>)].
2. Batchelor, G. K. (2000). *An introduction to fluid dynamics* (2nd pbk. ed.). Cambridge, U.K.: Cambridge University Press. ISBN 978-0-521-66396-0.
3. Hornik, Kurt; Tinchcombe, Maxwell; White, Halbert (1989-01-01). "Multilayer feedforward networks are universal approximators" (<https://www.sciencedirect.com/science/article/abs/pii/0893608089900208>). *Neural Networks*. **2** (5): 359–366. doi:10.1016/0893-6080(89)90020-8 (<https://doi.org/10.1016%2F0893-6080%2889%2990020-8>). ISSN 0893-6080 (<https://www.worldcat.org/issn/0893-6080>).
4. Arzani, Amirhossein; Dawson, Scott T. M. (2021). "Data-driven cardiovascular flow modelling: examples and opportunities" (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8086862>). *Journal of the Royal Society Interface*. **18** (175): 20200802. arXiv:2010.00131 (<https://arxiv.org/abs/2010.00131>). doi:10.1098/rsif.2020.0802 (<https://doi.org/10.1098%2Frsif.2020.0802>). PMC 8086862 (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8086862>). PMID 33561376 (<https://pubmed.ncbi.nlm.nih.gov/33561376>).
5. Arzani, Amirhossein; Wang, Jian-Xun; D'Souza, Roshan M. (2021-06-07). "Uncovering near-wall blood flow from sparse data with physics-informed neural networks". *Physics of Fluids*. **33** (7): 071905. arXiv:2104.08249 (<https://arxiv.org/abs/2104.08249>). Bibcode:2021PhFl...33g1905A (<https://ui.adsabs.harvard.edu/abs/2021PhFl...33g1905A>). doi:10.1063/5.0055600 (<https://doi.org/10.1063%2F5.0055600>). S2CID 233289904 (<https://api.semanticscholar.org/CorpusID:233289904>).

6. Raissi, Maziar; Perdikaris, Paris; Karniadakis, George Em (2019-02-01). "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations" (<https://www.sciencedirect.com/science/article/pii/S0021999118307125>). *Journal of Computational Physics*. **378**: 686–707. Bibcode:2019JCoPh.378..686R (<https://ui.adsabs.harvard.edu/abs/2019JCoPh.378..686R>). doi:10.1016/j.jcp.2018.10.045 (<https://doi.org/10.1016%2Fj.jcp.2018.10.045>). ISSN 0021-9991 (<https://www.worldcat.org/issn/0021-9991>). OSTI 1595805 (<https://www.osti.gov/biblio/1595805>). S2CID 57379996 (<https://api.semanticscholar.org/CorpusID:57379996>).
7. Markidis, Stefano (2021-03-11). "Physics-Informed Deep-Learning for Scientific Computing". arXiv:2103.09655 (<https://arxiv.org/abs/2103.09655>) [math.NA (<https://arxiv.org/archive/math.NA>)].
8. Baydin, Atilim Gunes; Pearlmutter, Barak A.; Radul, Alexey Andreyevich; Siskind, Jeffrey Mark (2018-02-05). "Automatic differentiation in machine learning: a survey". arXiv:1502.05767 (<https://arxiv.org/abs/1502.05767>) [cs.SC (<https://arxiv.org/archive/cs.SC>)].
9. Raissi, Maziar; Yazdani, Alireza; Karniadakis, George Em (2018-08-13). "Hidden Fluid Mechanics: A Navier–Stokes Informed Deep Learning Framework for Assimilating Flow Visualization Data". arXiv:1808.04327 (<https://arxiv.org/abs/1808.04327>) [cs.CE (<https://arxiv.org/archive/cs.CE>)].
10. Raissi, Maziar; Yazdani, Alireza; Karniadakis, George Em (2020-02-28). "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations" (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7219083>). *Science*. **367** (6481): 1026–1030. Bibcode:2020Sci...367.1026R (<https://ui.adsabs.harvard.edu/abs/2020Sci...367.1026R>). doi:10.1126/science.aaw4741 (<https://doi.org/10.1126%2Fscience.aaw4741>). ISSN 0036-8075 (<https://www.worldcat.org/issn/0036-8075>). PMC 7219083 (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7219083>). PMID 32001523 (<https://pubmed.ncbi.nlm.nih.gov/32001523>).
11. Mishra, Siddhartha; Molinaro, Roberto (2021-04-01). "Estimates on the generalization error of Physics Informed Neural Networks (PINNs) for approximating a class of inverse problems for PDEs". arXiv:2007.01138 (<https://arxiv.org/abs/2007.01138>) [math.NA (<https://arxiv.org/archive/math.NA>)].
12. Ryck, Tim De; Jagtap, Ameya D.; Mishra, Siddhartha (2022). "Error estimates for physics informed neural networks approximating the Navier–Stokes equations". arXiv:2203.09346 (<https://arxiv.org/abs/2203.09346>) [math.NA (<https://arxiv.org/archive/math.NA>)].
13. Dwivedi, Vikas; Parashar, Nishant; Srinivasan, Balaji (2021-01-08). "Distributed learning machines for solving forward and inverse problems in partial differential equations" (<https://www.sciencedirect.com/science/article/pii/S0925231220314090>). *Neurocomputing*. **420**: 299–316. doi:10.1016/j.neucom.2020.09.006 (<https://doi.org/10.1016%2Fj.neucom.2020.09.006>). ISSN 0925-2312 (<https://www.worldcat.org/issn/0925-2312>).
14. Rout, S., Dwivedi, V., & Srinivasan, B. (2019). Numerical Approximation in CFD Problems Using Physics Informed Machine Learning. *arXiv preprint arXiv:2111.02987*.
15. Dwivedi, Vikas; Parashar, Nishant; Srinivasan, Balaji (2021-01-08). "Distributed learning machines for solving forward and inverse problems in partial differential equations" (<https://www.sciencedirect.com/science/article/pii/S0925231220314090>). *Neurocomputing*. **420**: 299–316. doi:10.1016/j.neucom.2020.09.006 (<https://doi.org/10.1016%2Fj.neucom.2020.09.006>). ISSN 0925-2312 (<https://www.worldcat.org/issn/0925-2312>).

16. Jagtap, Ameya D.; Karniadakis, George Em (2020). "Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations". *Communications in Computational Physics*. **28** (5): 2002–2041. Bibcode:2020CCoPh..28.2002K (<https://ui.adsabs.harvard.edu/abs/2020CCoPh..28.2002K>). doi:10.4208/cicp.OA-2020-0164 (<https://doi.org/10.4208%2Fcicp.OA-2020-0164>). S2CID 229083388 (<https://api.semanticscholar.org/CorpusID:229083388>).
17. Jagtap, Ameya D.; Kharazmi, Ehsan; Karniadakis, George Em (2020). "Conservative Physics-Informed Neural Networks on Discrete Domains for Conservation Laws: Applications to forward and inverse problems" (<https://doi.org/10.1016%2Fj.cma.2020.113028>). *Computer Methods in Applied Mechanics and Engineering*. **365**: 113028. Bibcode:2020CMAME.365k3028J (<https://ui.adsabs.harvard.edu/abs/2020CMAME.365k3028J>). doi:10.1016/j.cma.2020.113028 (<https://doi.org/10.1016%2Fj.cma.2020.113028>). S2CID 216333271 (<https://api.semanticscholar.org/CorpusID:216333271>).
18. Leake, Carl; Mortari, Daniele (12 March 2020). "Deep Theory of Functional Connections: A New Method for Estimating the Solutions of Partial Differential Equations" (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7259480>). *Machine Learning and Knowledge Extraction*. **2** (1): 37–55. doi:10.3390/make2010004 (<https://doi.org/10.3390%2Fmake2010004>). PMC 7259480 (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7259480>). PMID 32478283 (<https://pubmed.ncbi.nlm.nih.gov/32478283>).
19. Schiassi, Enrico; Furfaro, Roberto; Leake, Carl; De Florio, Mario; Johnston, Hunter; Mortari, Daniele (October 2021). "Extreme theory of functional connections: A fast physics-informed neural network method for solving ordinary and partial differential equations". *Neurocomputing*. **457**: 334–356. doi:10.1016/j.neucom.2021.06.015 (<https://doi.org/10.1016%2Fj.neucom.2021.06.015>). S2CID 236290147 (<https://api.semanticscholar.org/CorpusID:236290147>).
20. Schiassi, Enrico; De Florio, Mario; Ganapol, Barry D.; Picca, Paolo; Furfaro, Roberto (March 2022). "Physics-informed neural networks for the point kinetics equations for nuclear reactor dynamics". *Annals of Nuclear Energy*. **167**: 108833. doi:10.1016/j.anucene.2021.108833 (<https://doi.org/10.1016%2Fj.anucene.2021.108833>). S2CID 244913655 (<https://api.semanticscholar.org/CorpusID:244913655>).
21. Schiassi, Enrico; D'Ambrosio, Andrea; Drozd, Kristofer; Curti, Fabio; Furfaro, Roberto (4 January 2022). "Physics-Informed Neural Networks for Optimal Planar Orbit Transfers". *Journal of Spacecraft and Rockets*. **59** (3): 834–849. Bibcode:2022JSpRo..59..834S (<https://ui.adsabs.harvard.edu/abs/2022JSpRo..59..834S>). doi:10.2514/1.A35138 (<https://doi.org/10.2514%2F1.A35138>). S2CID 245725265 (<https://api.semanticscholar.org/CorpusID:245725265>).
22. De Florio, Mario; Schiassi, Enrico; Ganapol, Barry D.; Furfaro, Roberto (April 2021). "Physics-informed neural networks for rarefied-gas dynamics: Thermal creep flow in the Bhatnagar–Gross–Krook approximation". *Physics of Fluids*. **33** (4): 047110. Bibcode:2021PhFl...33d7110D (<https://ui.adsabs.harvard.edu/abs/2021PhFl...33d7110D>). doi:10.1063/5.0046181 (<https://doi.org/10.1063%2F5.0046181>). S2CID 234852360 (<https://api.semanticscholar.org/CorpusID:234852360>).

23. Kashefi, Ali; Mukerji, Tapan (2022). "Physics-informed PointNet: A deep learning solver for steady-state incompressible flows and thermal fields on multiple sets of irregular geometries". *Journal of Computational Physics*. **468**: 111510. arXiv:2202.05476 (<https://arxiv.org/abs/2202.05476>). Bibcode:2022JCoPh.46811510K (<https://ui.adsabs.harvard.edu/abs/2022JCoPh.46811510K>). doi:10.1016/j.jcp.2022.111510 (<https://doi.org/10.1016%2Fj.jcp.2022.111510>). S2CID 246823515 (<https://api.semanticscholar.org/CorpusID:246823515>).
24. Qi, Charles; Su, Hao; Mo, Kaichun; Guibas, Leonidas (2017). "Pointnet: Deep learning on point sets for 3d classification and segmentation" (https://openaccess.thecvf.com/content_cvpr_2017/papers/Qi_PointNet_Deep_Learning_CVPR_2017_paper.pdf) (PDF). *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*: 652–660. arXiv:1612.00593 (<https://arxiv.org/abs/1612.00593>).
25. Aislan Antonelo, Eric; Camponogara, Eduardo; Seman, Laio; Jordanou, Jean Panaioti; de Souza, Eduardo Rehbein; Hubner, Jomi (2023-02-02). "Physics-Informed Neural Nets for Control of Dynamical Systems" (<https://papers.ssrn.com/abstract=4346399>). Rochester, NY. doi:10.2139/ssrn.4346399 (<https://doi.org/10.2139%2Fssrn.4346399>).
26. Wang, Sifan; Teng, Yujun; Perdikaris, Paris (2020-01-13). "Understanding and mitigating gradient pathologies in physics-informed neural networks". arXiv:2001.04536 (<https://arxiv.org/abs/2001.04536>) [cs.LG (<https://arxiv.org/archive/cs.LG>)].

External links

- PINN (<https://github.com/maziarraissi/PINNs/>) – repository to implement physics-informed neural network in Python
 - XPINN (<https://github.com/AmeyaJagtap/XPINNs/>) – repository to implement extended physics-informed neural network (XPINN) in Python
-

Retrieved from "https://en.wikipedia.org/w/index.php?title=Physics-informed_neural_networks&oldid=1144883214"