Journal of Nonlinear Modeling and Analysis
Volume 6, Number 2, June 2024, 485–513
http://jnma-online.com
DOI:10.12150/jnma.2024.485

Meta-Learning-Based Physics-Informed Neural Network: Numerical Simulations of Initial Value Problems of Nonlinear Dynamical Systems Without Labeled Data and Correlation Analyses

Worrawat Duanyai[1], Weon Keun Song[2,†], Thanadol Chitthamlerd[3] and Girish Kumar[4]

Abstract There are several main challenges in solving nonlinear differential equations with artificial neural networks (ANNs), such as a nonlinear system's sensitivity to its initial values, discretization, and strategies for incorporating physics-based information into ANNs. As for the first issue, this paper addresses the initial value problems of nonlinear dynamical systems (a Duffing oscillator and a Burger's equation), which cause large global truncation errors in sub-domains with a significant reduction in the influence of initial constraints, using meta-learning-based physics-informed neural networks (MPINNs). The MPINNs with dual learners outperform physics-informed neural networks with a single learner (no fine reinitialization capability). As a result, the former approach improves solution convergence by 98.83% in the sub-time domain (III) of a Duffing oscillator, and by 85.89% at $t = 45$ in a Burger's equation problem, compared to the latter one. Model accuracy is highly dependent on the adaptability of the initial parameters in the first hidden layers of the meta-models. From correlation analyses, it is obvious that the parameters become less (the Duffing oscillator) or more (the Burger's equation) correlated during fine reinitialization, as the update manner differs or is similar to the one used in pre-initialization. In the first example, the MPINN achieves both the mitigation of model sensitivity to its output and the improvement of model accuracy. Conversely, the second example shows that the proposed approach is not enough to solve both issues simultaneously, as increased model sensitivity to its output leads to higher model accuracy. The application of transfer learning reduces the number of iterative pre-meta-trainings.

[†]the corresponding author.
Email address: worrawatduanyai@gmail.com(W. Duanyai), bauman98@naver.com(W K. Song), thanadolps@gmail.com(T. Chitthamlerd), girish.kumar154@gmail.com(G. Kumar)
[1]Department of Robotic and Computational Intelligence System, School of Engineering, King Mongkut's Institute of Technology Ladkrabang, Chalong Krung 1 Rd.,10520 Bangkok, Thailand
[2]Department of Robotic and AI Engineering, School of Engineering, King Mongkut's Institute of Technology Ladkrabang, Chalong Krung 1, 10520 Bangkok, Thailand
[3]Department of Computer Engineering, Chulalongkorn University, Phaya Thai Rd, Wang Mai, 10330 Bangkok, Thailand
[4]Department of Mechanical Engineering, Delhi Technological University, Shahbad Daulatpur, Bawana Road, 110042 Delhi, India

Keywords Physics-informed neural network, meta-learning, fine reinitialization, transfer learning, initial value problem, nonlinear dynamical systems, correlation analyses

MSC(2010) 65L99, 65M99.

Nomenclature

Symbol	Definition
$\boldsymbol{a}^{[l]}$, $a_n^{[l]}$	Output vector of the l-th hidden layer and the n-th component of $\boldsymbol{a}^{[l]}$
$\boldsymbol{b}^{[l]}$	Bias vector of the l-th hidden layer
c	Constant parameter
cn, dn, sn	Jacobi elliptic functions
d	Constant parameter
$\boldsymbol{d}_k$	Search direction vector at the k-th iteration
$\boldsymbol{D}_{qu}^{(i)}$, $\boldsymbol{D}_{su}^{(i)}$	Query and support datasets, respectively, for the i-th query task and the i-th support one.
$\boldsymbol{D}_{te}, \boldsymbol{D}_{tr}$	Test and training datasets, respectively
f	Constant parameter
$f(\boldsymbol{u})$	Function value at $\boldsymbol{u}$
$f_{\boldsymbol{\theta}}(\boldsymbol{x})$	PINN
$f_{\boldsymbol{\theta}}^m(\boldsymbol{x})$	MPINN
$f_{\boldsymbol{\theta}^{(i)}}^m(\boldsymbol{x}_{su}^{(i)})$	A model for the i-th support task (it is regarded as the i-th support task)
$f_{\boldsymbol{\theta}^{(i)},'}^m(\boldsymbol{x}_{qu}^{(i)})$	A model for the i-th query task (it is regarded as the i-th query task)
$f_{\boldsymbol{\theta}^*}^m(\boldsymbol{x}_{te})$	MPINN evaluated on $\boldsymbol{D}_{te}$
$\boldsymbol{g}_k$	Gradient difference vector at the k-th iteration
$\boldsymbol{H}^{(I)}$, $\boldsymbol{H}_k$	Initial Hessian matrix and Hessian matrix at the k-th iteration
$iter$, $iter^m$	Iterative numbers, respectively, for the PINN (or each inner learner of the MPINN) and the outer learner of the MPINN
$iter^{MAX}$, $iter^{mMAX}$	Maximum iterative numbers, respectively, for the PINN (or each inner learner of the MPINN) and the outer learner of the MPINN.
k	Constant parameter
$L(\boldsymbol{\theta},\ \boldsymbol{D}_{tr})$	Total PINN loss
$L^{(i)}(\boldsymbol{\theta}^{(i)}, \boldsymbol{D}_{su}^{(i)})$, $L^{(i)}(\boldsymbol{\theta}^{(i),'}, \boldsymbol{D}_{qu}^{(i)})$	Support and query losses, respectively, for the i-th task
$\|\|\nabla L(\boldsymbol{\theta}, \boldsymbol{D}_{tr})\|\|$	Norm of the projected gradient of $L(\boldsymbol{\theta}, \boldsymbol{D}_{tr})$
m	Constant parameter
MSE_f, MSE_I, MSE_B, MSE_d	Losses, respectively, of residual, initial constraint, boundary constraint, and labeled data
N_d, N_B, N_f, N_I	Number of labeled data for MSE_d and number of training samples, respectively, for MSE_B, MSE_f and MSE_I
NT	Number of test samples
NQ, NS	Numbers, respectively, of query and support tasks
t, $\triangle t$	Time coordinate and time increment
u, u_n	A single output of $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ and a discrete solution of u at the n-th sampling point
$\boldsymbol{u}$	Output vector of $f_{\boldsymbol{\theta}}(\boldsymbol{x})$
$\bar{u}$, $\bar{u}_n$	Analytical (exact) solution and a discrete solution of $\bar{u}$ at the n-th sampling point
$\boldsymbol{u}_{su}^{(i)}$, $\boldsymbol{u}_{qu}^{(i),'}$, $\boldsymbol{u}_{te}$	Output vectors, respectively, evaluated by $f_{\boldsymbol{\theta}^{(i)}}^m(\boldsymbol{x}_{su}^{(i)})$, $f_{\boldsymbol{\theta}^{(i)},'}^m(\boldsymbol{x}_{qu}^{(i)})$, and $f_{\boldsymbol{\theta}^*}^m(\boldsymbol{x}_{te})$
$\bar{\boldsymbol{u}}_{su}^{(i)}$, $\bar{\boldsymbol{u}}_{qu}^{(i)}$, $\bar{\boldsymbol{u}}_{te}$	Actual output vectors, respectively, for $\boldsymbol{D}_{su}^{(i)}$, $\boldsymbol{D}_{qu}^{(i)}$, and $\boldsymbol{D}_{te}$ ($\bar{\boldsymbol{u}}_{su}^{(i)}$ and $\bar{\boldsymbol{u}}_{qu}^{(i)}$ require no labeled data)
D_{te}^n, ∂_x^n, ∂_t^n	n-th order differential operators
x	Space coordinate
$\boldsymbol{x}$	Input vector