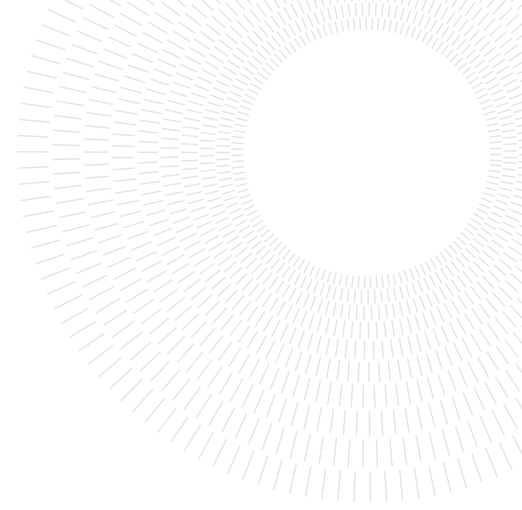




POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE



A fully-comprehensive library for Physics-Informed Deep Learning under uncertainty

Project Report

Advanced Programming for Scientific Computing

Authors:

Giulia Mescolini
Luca Sosta

Professor:

Prof. Luca Formaggia

Tutors:

Prof. Andrea Manzoni
Prof. Stefano Pagani

Academic Year:

2021-2022

Abstract: in this project, we developed a new library implementing several variants of the Bayesian Physics-Informed Neural Networks (B-PINNs) method, a framework to tackle physical problems with Neural Networks that are able to take into account information coming from partial differential equations and quantify the uncertainty around their prediction.

To implement this strategy, we designed from scratch a modular and flexible pipeline; in our design, indeed, on one hand we wanted to mirror the peculiarities of the network tasks involved in the B-PINN framework into the hierarchy of the classes, while on the other hand we aimed at proposing a structure on which a variety of functionalities for different tasks could be inserted.

In the proposed release, indeed, we provided different variants of the various aspects of the overall process, ranging from data management to training algorithms, among which we implemented Adam, Hamiltonian Monte Carlo, Variational Inference and Stein Variational Gradient Descent.

The library design is also predisposed to further extensions, because the blocks representing the different components that detach from the main skeleton are thought to be defined by few case-specific behaviors and not to have heavy constraints on the functionalities, as validated when implementing algorithms with distinct routines and structures involved or different problems.

Finally, we proposed a showcase of results to highlight the main features of method and library: algorithm comparison, introduction of PDE residuals into Neural Networks, portability to high dimension and quality of results that can be obtained with fine-tuning of model parameters.

Key-words: B-PINNs, Uncertainty Quantification, Scientific Learning

Contents

1	Introduction	4
1.1	Project Overview	4
1.2	Report Structure	5
2	Methods	6
2.1	An overview on Neural Networks	6
2.1.1	Core Structure	6
2.1.2	Backpropagation	8
2.1.3	Optimizers	9
2.1.4	Training Neural Networks	11
2.2	Physics Informed Neural Networks	12
2.2.1	Structure of a PINN	12
2.2.2	Optimization Problems	13
2.2.3	Dataset for PINNs	14
2.3	Bayesian Neural Networks	15
2.3.1	Likelihood Distribution	16
2.3.2	Prior Distribution	16
2.3.3	Posterior Distribution	17
2.4	Bayesian Physics Informed Neural Networks	17
2.4.1	Physical Likelihood	18
2.4.2	Prior for Inverse Problems	18
3	Algorithms	19
3.1	Hamiltonian Monte Carlo	19
3.1.1	Theoretical Foundations	20
3.1.2	Algorithm	21
3.1.3	Parameters' choice	22
3.2	Variational Inference	22
3.2.1	Theoretical Foundations	22
3.2.2	Algorithm	23
3.2.3	Parameters' choice	24
3.3	Stein Variational Gradient Descent	24
3.3.1	Theoretical Foundations	24
3.3.2	Algorithm	25
3.3.3	Parameters' choice	25
4	Code Overview	26
4.1	Working Environment	26
4.1.1	Interpreter	26
4.1.2	Virtual Environment	26
4.2	Libraries	26
4.2.1	Built-in Packages	26
4.2.2	External Packages	27
4.2.3	TensorFlow	27
4.3	Object Oriented Features	28
4.3.1	Inheritance	28
4.3.2	Abstract Base Class	29
4.3.3	Dataclass	30
4.3.4	Iterators	30
4.3.5	Property	31
4.4	Repository Structure	31
4.4.1	Configuration - <code>config</code> Folder	31
4.4.2	Datasets - <code>data</code> Folder	33
4.4.3	Outputs - <code>outs</code> Folder	33

5	Source Code	35
5.1	Main Files	36
5.1.1	Main Pipeline	36
5.2	Parameters Handling	36
5.2.1	Configuration files	36
5.2.2	Command line arguments	37
5.2.3	Param object	37
5.3	Data Generation	37
5.3.1	Domain creation	37
5.3.2	Multi-domain creation	38
5.3.3	Boundary Points	38
5.3.4	Data Configuration	39
5.4	Pre-Processing	39
5.4.1	Dataset creation	40
5.4.2	Data normalization	40
5.4.3	Data Batching	40
5.5	Equations	41
5.5.1	Operators	41
5.5.2	Abstract Equation	42
5.5.3	Problems Implemented	42
5.6	Model	43
5.6.1	Parameters object - Theta	43
5.6.2	Parameters and forward pass - CoreNN	45
5.6.3	Physics enforcement - PhysNN	46
5.6.4	Training functionalities - LossNN	46
5.6.5	Prediction phase - PredNN	47
5.6.6	Final wrapper - BayesNN	48
5.7	Optimizers	49
5.7.1	Training interface	49
5.7.2	Algorithm abstract class	49
5.7.3	Adam optimizer	50
5.7.4	Hamiltonian Monte Carlo	50
5.7.5	Variational Inference	51
5.7.6	Stein Variational Gradient Descent	51
5.8	Postprocessing	52
5.8.1	Storage	52
5.8.2	Plotter	53
6	Results	54
6.1	Regression Problem	54
6.1.1	Adam	54
6.1.2	HMC	55
6.1.3	SVGD	56
6.1.4	VI	56
6.2	Damped Harmonic Oscillator Problem	57
6.2.1	NN vs PINN	57
6.3	Multi-dimensional domain	57
6.3.1	Adam 2D cosine	57
6.4	HMC Showcase	58
6.4.1	Regression Problem - sine	58
6.4.2	Oscillator Problem	59
6.4.3	Poisson/Laplace Problem - cosine	60
6.4.4	Poisson/Laplace Problem - sine	61
7	Conclusions	62
A	Installation Guide	63