CEMRACS 2023 mini-project proposal

Title: 1-Lipschitz neural networks for error control in function approximation

Contact: Paul Novello, Clément Gauchy
Institution or company: DEEL (IRT Saint Exupéry, ANITI), CEA
Email: paul.novello@irt-saintexupery.com, clement.gauchy@cea.fr
Phone number: +33 6 28 34 35 58

1. Context

Neural networks are increasingly used in the context of scientific computing. Indeed, once trained, they can approximate highly complex, non-linear, and high dimensional functions for dramatically less time than traditional simulation codes based on finite-differences methods. However, unlike traditional simulation, whose error can be controlled, neural networks are statistical, data-driven models, so no approximation error guarantee can be naturally provided. This limitation prevents neural networks from being safely used on par with finite elements-based simulation codes in scientific computing.

Several deep-learning applications, such as adversarial robustness [5] or optimal transport [6], have recently stimulated research about the construction of 1-Lipschitz neural networks [1,2]. The rigid constraints imposed to enforce the 1-Lipschitz property lead to networks of remarkable regularity, and the knowledge of their Lipschitz constant enable certificates for perturbation robustness [2,5]. These characteristics of 1-Lipschitz networks (regularity and certifiability) make them particularly appealing for applications in scientific computing.

At DEEL, a research lab that is part of ANITI and IRT Saint Exupery, an important line of research is focused on 1-Lipschitz networks and their application to vision tasks [2,5,7,8]. This project aims to study these networks for approximating numerical solutions of PDEs, intensively used at the CEA.

2. Description and objectives

This project aims to study 1-Lipschitz networks for function approximation with an emphasis on the analysis of their benefits in terms of regularity and error control. The project's first objective is to study and improve their capacity to approximate complex functions that are solutions to PDEs. The second objective is to theoretically derive generalization bounds based on the Lipschitz property of the neural networks and to verify them empirically. Ultimately, the question that the project is intended to help answer is "are 1-Lipschitz networks capable of accurately approximating PDE solutions together with providing relevant error guarantees?". A positive answer to this question would further unlock the use of Deep Learning in Scientific Computing.

3. Proposed methodology

The methodologies designed during the project will be empirically evaluated on regression of numerical solutions of simple PDE resolution problems.

1) Due to constraints imposed during training and the value of their Lipschitz constant, 1-Lipschitz networks are known to be less expressive than unconstrained networks. Therefore, the project's first step will be to obtain a decent approximation error with Lipschitz networks. Several ways can be suggested.

- To make the network >1 Lipschitz:
    - Removing the 1-Lipschitz constraint in the last linear layer of the neural network.
    - Multiplying the output of the network component-wise by a trainable vector.
    - Multiplying the output of the network component-wise by a vector which is the output of an auxiliary neural network.

- To make the network more accurate/expressive:
    - Add a Physics-Informed regularization term [3] in the loss of the network.
    - Construct 1-Lipschitz Neural Operator [4,9].
    - Find applications where the Lipschitz constraint of the neural network can be seen as an implicit bias.

2) Then, the regularity of the obtained neural networks will be empirically and theoretically assessed:

- **Theoretically:** Construction of a generalization bound on max error based on the maximum variation (according to the Lip constant) of the neural network between points of the training set, under hypotheses on the function to approximate. This part is theoretical and open-ended, but several lines are possible, ranging from a simple bound in 1-2 dimensions from a regular grid that can be derived easily to more complex bounds in N dimensions (exploratory, but some ideas will be presented to the participants).
- **Empirically**: Evaluation of max error with an intensive sampling of the input space. Validation of the theoretical bound and comparison with a regular neural network.

The time spent on the project can be divided between 1) and 2) according to the participant's preferences and progression.

3) Software requirements

The project will be conducted in Python. Required packages are (not exhaustive and definitive list):

py-pde (zwicker-group/py-pde: Python package for solving partial differential equations using finite differences. (github.com)): A python package to simulate solutions of simple PDEs.
PDEBench (pdebench/PDEBench: PDEBench: An Extensive Benchmark for Scientific Machine Learning (github.com))
deel-lip (deel-ai/deel-lip: Tensorflow 2 implementation of k-Lipschitz layers. (github.com)): A python package, based on tensorflow, to construct 1-Lipschitz networks. A Pytorch version is available (but less mature) at deel-ai/deel-torchlip (github.com)

4) References

[1] Sorting out Lipschitz function approximation, Anil et al, ICML 2019
[2] Pay attention to your loss: understanding misconceptions about 1-Lipschitz neural networks, Bethune et al, NeurIPS 2022
[3] Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Raissi et al, Journal of Computational Physics, 2019
[4] Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, Nature Machine Intelligence, 2021
[5] Achieving robustness in classification using optimal transport with hinge regularization, Serrurier et al. CVPR 2021
[6] Wasserstein GAN, Arjovsky et al., ICML 2017
[7] GAN estimation of Lipschitz optimal transport maps, Gonzalez-Sanz et al., 2022
[8] When adversarial attacks become interpretable counterfactual explanations, Serrurier et al., 2022
[9] Physics-Informed Deep Neural Operator Networks, Goswami, 2022