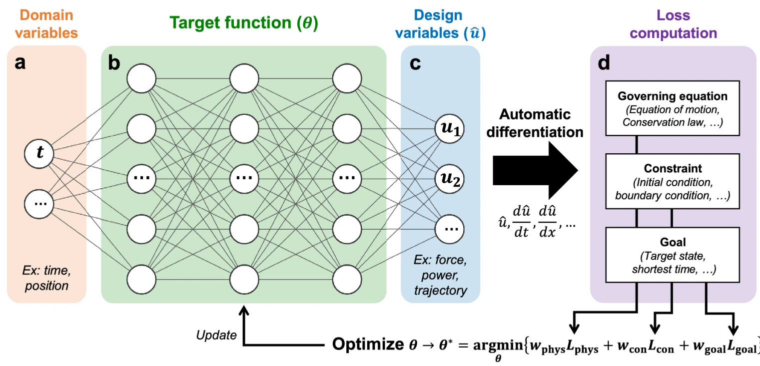
036049: Applied ML4SE, Winter 2023/2024
Final Project Details

Due: May 7, 2024



Introduction

The focus of this class has been on machine learning with an eye towards applications in science and engineering. This resulted in our focus on supervised learning (mostly) and deep neural networks. Specifically, we have worked our way towards studying physics-informed neural networks (PINNs). PINNs are part of a broader class of models related to scientific machine learning (SciML). The goal is guide the expressive power of neural networks by training them subject to physics-related constraints often in the form of partial differential equations (PDEs) related to the problem domain especially when faced with limited training data.

In this final project, you are being asked to consider *implementation and/or modification of a PINNs related approach to a problem of your interest.* The problem may come from one of the main sub-disciplines of mechanical engineering such as fluid dynamics, heat transfer, combustion, solid mechanics, or materials, or any other field in science and engineering. Ideally, your project should be loosely based on an published archival journal publication from a reputable scientific or engineering journal. It could involve the modification of an existing PINNs-related code base that you obtained from `GitHub` e.g. this one is tied to this very nice review article here or from one of the sample PINNs codes we provided to you in class. Another option is to pursue a more traditional deep neural network (with PINNs) but this would require sufficient labeled training data for your application which is not always readily available or easy to train. You will develop your code and results in Python using either VSCode or Google Colab.

If, for some reason, you prefer to work on project not involving PINNs, here are some comments/-suggestions/issues to consider. First, you must, at least, work with a deep neural network (DNN). You could consider a different architecture (even one we have not discussed yet) such as a convolutional neural network, or a variational autoencoder, etc. You should be aware that avoiding PINNs, suggests you will need labeled training data. For engineering applications such as fluid/thermal science, or mechanics/materials, this may be hard to come by. Part of the draw of PINNs is that you don't need labeled training data as you train the network using initial/boundary conditions and governing equations (ODEs/PDEs) for your problem. You may also consider going beyond PINNs to consider neural operators and one of the items (25) listed below.

General project topic ideas

Here are a few of my general suggestions for PINNs related topics:

1. PINNs for solving the incompressible Navier-Stokes equations for low-speed fluid flow e.g. lid-driven cavity, flow over a cylinder, etc.
2. PINNs for solving the compressible Euler equations for high-speed inviscid flow e.g. 1D shock tube problem, 2D Riemann problem, etc.
3. PINNs for solving the compressible Navier-Stokes equations for high-speed viscous flow e.g. viscous shock tube problem, etc.
4. PINNs for heat transfer e.g. heat conduction equation in multiple dimensions
5. PINNs for solid mechanics e.g. linear elasticity equation

Here are some suggestions from ChatGPT:

1. Fluid Dynamics:
   - Using convolutional neural networks (CNNs) to predict flow patterns and velocities in complex geometries.
   - Implementing physics-informed neural networks (PINNs) to solve Navier-Stokes equations for fluid flow.
2. Heat transfer:
   - Predicting heat transfer coefficients using deep neural networks.
   - Developing a model using CNNs to predict temperature distributions in thermal systems.
3. Mechanics
   - Using recurrent neural networks (RNNs) to predict stress and strain distributions in mechanical components.
   - Implementing a physics-informed neural network for structural health monitoring.
4. Materials
   - Predicting material properties (e.g., strength, elasticity) using deep neural networks.
   - Developing a model to classify material types based on microstructure images using CNNs.
5. PINNs
   - Comparing the performance of PINNs with traditional numerical solvers in solving mechanical engineering problems.
   - Investigating the impact of different loss functions in PINNs for fluid dynamics simulations.

6. General Deep Learning Approaches
   - Image-based classification of mechanical components or defects using CNNs.
   - Time-series analysis of mechanical systems using RNNs or transformer models.
7. Interdisciplinary Projects
   - Collaborating with students from other disciplines (e.g., materials science, physics) to tackle complex problems using deep learning.
   - Integrating sensor data with neural networks for real-time monitoring and control of mechanical systems.

These projects can be tailored to the specific interests and expertise of the students, and they provide a good balance between using deep neural networks and related architectures for various mechanical engineering applications.

Recent relevant references for project ideas

Below is a hyperlinked list of recent references related to PINNs and its application to science and engineering obtained by simply Googling. Scan the list and see if anything catches your eye. You can Google too for other ideas/references. The best papers have GitHub repositories where you can download and run their code and use their data. Obviously for your project you must go beyond simply downloading and running someone else's code but it is not a bad place to start.

1. SenseNet: A Physics-Informed Deep Learning Model for Shape Sensing
2. Physics-Informed Neural Networks for Low Reynolds Number Flows over Cylinder
3. PINNs for heat transfer
4. A physics-informed deep learning method for solving direct and inverse heat conduction problems of materials
5. Analyses of internal structures and defects in materials using physics-informed neural networks
6. Physics-informed neural networks approach for 1D and 2D Gray-Scott systems
7. M-PINN: A mesh-based physics-informed neural network for linear elastic problems in solid mechanics
8. A Gentle Introduction to Physics-Informed Neural Networks, with Applications in Static Rod and Beam Problems
9. An Expert's Guide to Training Physics-informed Neural Networks
10. A Review of Physics-Informed Machine Learning in Fluid Mechanics
11. Physics-informed machine learning

12. Physics-informed neural networks for high-speed flows
13. Physics-informed deep learning for simultaneous surrogate modeling and PDE-constrained optimization of an airfoil geometry
14. Physics-informed neural networks for predicting gas flow dynamics and unknown parameters in diesel engines
15. Stiff-PINN: Physics-Informed Neural Network for Stiff Chemical Kinetics
16. Learning stiff chemical kinetics using extended deep neural operators
17. Physics-informed neural networks for modeling physiological time series for cuffless blood pressure estimation
18. Physics-informed neural networks for solving Reynolds-averaged Navier-Stokes equations
19. Physics-informed neural networks for solving time-dependent mode-resolved phonon Boltzmann transport equation
20. Solving multi-material problems in solid mechanics using physics-informed neural networks based on domain decomposition technology
21. Scientific Machine Learning Through Physics-Informed Neural Networks: Where we are and What is Next
22. Solving real-world optimization tasks using physics-informed neural computing
23. Discontinuity Computing Using Physics-Informed Neural Networks
24. Machine Learning in Fluid Dynamics
25. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators

Submission Guidelines

In addition to performing the computations you are tasked to prepare a detailed report including the following sections:

- **Abstract** (1 paragraph describing what you did and what you found)
- **Introduction** (1-2 pages providing background and a brief literature review motivating your project)
- **Methods** (2-3 pages on the mathematical/computational model)
- **Problem description** (1-2 pages clearly describing the problem you are trying to solve including schematics/figures)
- **Results and discussion** (3-5 pages presenting (graphically or otherwise), discussing, and analyzing the results you obtained and their significance).

- **Conclusions** (1 paragraph summarizing what you did, what you found, and why it is relevant)
- **References** (proper bibliography of any/all references cited in your report)

Your report should be prepared with your favorite word processor (here are templates for LaTeX and also Word and LaTeX(outside of overleaf) versions. Another option is here.

You will also share your working code(s) as either a Python file(s) or better as a Jupyter Notebook. We will try to run your code to reproduce your results for verification purposes.

Timeline and deliverables

Below is a timeline and deliverable list. PDFs of your deliverables must be uploaded to the Moodle page.

- March 12, 2024: A 1-2 page proposal for your topic and plan must be submitted for approval. This should include the specific paper you plan to follow and also the `GitHub` site or code you plan to use/modify.
- April 9, 2024: A preliminary 3-5 page progress report must be submitted including details and any preliminary results.
- May 7, 2024: This is the final project report due data.