Project Proposals

Damien Beecroft

August 2022

Introduction

How does one appropriately combine the vast wealth of theory from numerical analysis and dynamical systems with machine learning and data science to most efficiently and accurately solve differential equations? This is a complex problem whose answer depends heavily upon the equation, the available data, and a multitude of other factors. None of the work on physics informed neural networks (PINNs) that I have seen leverages the vast wealth of existing numerical methods. Is there a way of combining the theory of these two worlds to invent algorithms that outperform numerical analysis or PINNs alone? If so, why and when can physics simulations benefit from machine learning? Furthermore, can the benefits of machine learning be made robust or does one need to cherry pick results in order to see improvement?

Project Ideas

Project One

The first and central project is to use a numerical method as the single fidelity solver for a multifidelity finite basis PINN (MFFBPINN). I will start by choosing a set of test problems and running numerical methods to ascertain their speed and accuracy. Then, I will take these numerical methods and make them the single fidelity solvers for MFFBPINNs. I will then do a comparison between the classical numerical methods and the MFFBPINNs. Can machine learning push the numerical method past the round off error that occurs due to the inherent ill conditioning of numerical methods? How do the timings of the methods compare? Would it be simply more beneficial to make a finer grid as opposed to adding a neural network? Are the results different if the problem domain is not regular and grid collocation is no longer an easy task? What if we have access to solution data or theoretical knowledge of fixed point? Can MFFBPINNs correct for the biases in numerical solvers e.g. the artificial viscosity introduced in some finite volume methods for the inviscid Burger's equation?

Project Two

After the PINN experiments are done I want to repeat the studies from project one with multifidelity finite basis DeepONets. DeepONets do not work well alone. I wonder whether the aide of numerical analysis can make DeepONets viable. This is the very important experiment. Having to retrain a network for every set of initial conditions is unreasonable in a wide range of applications.

Project Three

I think it would be interesting to look at the loss landscape of the MFFBPINNs with numerical solvers. We know that at the beginning of training you are within a certain error ϵ of the true solution. This means that evaluating a linear or quadratic approximation of the neural network at initialization would likely give one a good approximation of the loss landscape at the minima. Furthermore, there is something that has struck me as a bit strange about the PINN training process. In numerical methods the density of grid points is paramount to successful convergence. However, density of collocation points in batches for PINNs is not–at least in my experience–treated with the same importance. When one creates a batch of collocation points I believe they should be sampled at or above the Nyquist frequency of the highest frequency mode in the solution. I am curious to see whether my intuition can be tested by analyzing the loss landscape. It may also be interesting to analyze the convergence of MFFBPINNs with numerical solvers through a combination of numerical solvers and the neural tangent kernel.