Graduate Research Plan Statement for Deep Learning of Partial Differential Equations

Before beginning graduate school, I had planned to focus my research on computational efficiency. In the past several years, deep learning has come to dominate this field and has had extraordinary implications in solving wide range of practical problems in medical imaging, signal processing, computer vision, remote sensing, electromagnetism and more[4]. More recently, the use of machine learning has allowed for highly accurate approximations of solutions of computationally costly partial differential equations (PDEs) [4]. With my physics-focused background, I find this new area of research enticing as PDEs are used to describe many systems in physics. My proposed research will combine my background in physics with my skills developed in computational mathematics and numerical analysis. In this endeavor, I will contribute to the development of a more rigorous theory and understanding of convergence criteria of neural networks used to approximate solutions of PDEs, which is currently insufficiently explored. To achieve this, I will:

(1) Apply neural networks to find solutions to a variety of PDEs which are currently impractical to solve due to computational cost. I will experiment with several architectures to compare and contrast convergence rates.

(2) Develop further theory relating to the use of neural networks to approximate solutions of PDEs; particularly, I will explore the convergence criteria for a various classes of PDEs.

Intellectual Merit

As discussed in my personal statement, I have worked on a variety of research projects that rely on computational efficiency and robustness. My proposed project is no exception. I have experience with solving PDEs both analytically and using finite difference methods in several physics and mathematics courses; it became very routine for me to solve the Schroedinger, and wave equations with varying parameters, and conditions. This experience will serve as a gauge for solutions found using a neural network. Starting with simple PDEs and network architectures will serve as a proof-of-concept before attempting to solve more complicated problems. For example, the one-dimensional traveling wave: $\dot{u} + cx' = 0$. At first glance, this seems like a decent starting point for research; having only first order derivatives and knowing the actual solution embodies the idea of simplicity. However, during a naive attempt to learn the solution the using "Deep Galerkin Methods" (DGMs) published by Sirignano and Spiliopoulos, we find the network does not consistently converge to the desired solution, see Figure 1. These DGMs represent the solution of a PDE as a neural network that is trained to satisfy the differential operator, initial condition, and boundary conditions using stochastic gradient decent at randomly selected points [2]. Our *seemingly* non-converging example suggests one of the following: particular PDEs have some underlying properties that guarantee the network will converge to a solution (Sirignano and Spiliopoulos provide a theorem for the convergence of multi-layer feed-forward networks to approximate solutions of quasilinear parabolic PDEs) [2], or the use of more sophisticated algorithms or network architectures is paramount to finding solutions to certain PDEs. With this traveling wave serving as a potential counter-example, I find opportunity for further exploration into convergence criteria and algorithm optimization.

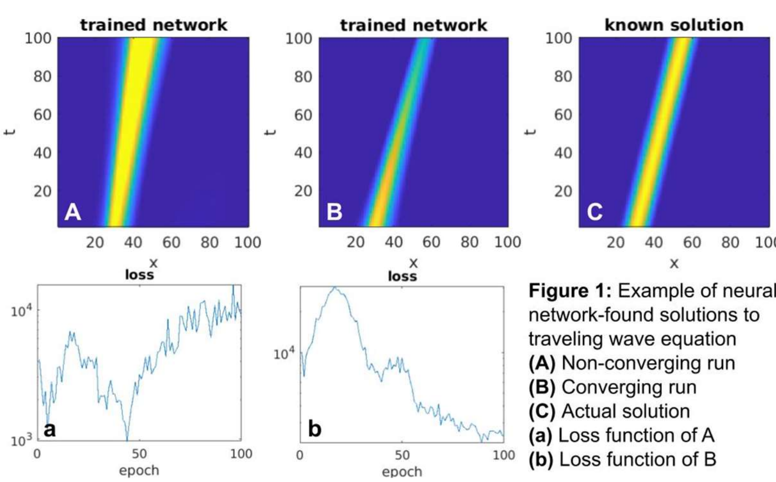

Figure 1: Example of neural network-found solutions to traveling wave equation
(A) Non-converging run
(B) Converging run
(C) Actual solution
(a) Loss function of A
(b) Loss function of B

There is also opportunity for further exploration into the use of prior knowledge of a given system to serve as a further constraint on solutions of PDEs found by a neural network [2]. For many problems in physics in particular, solutions to a given PDE must also satisfy conservation laws. One example, provided by Raissi, Perdikaris, and Karniadakis, constrains solutions of a flow problem to found solutions that do not violate conservation of mass [2]. This use of prior knowledge improves computation efficiency by restricting the space of possible solutions found by a neural network. These techniques can be applied to a wide range of other practical problems, including the example seen in Figure 1 as the wave seems to spread or lose amplitude, which it should not do. The use of prior knowledge to impose constraints on a PDE solving neural network for more computational efficiency is an idea that is very familiar to me with my background in studying physical systems; in mechanics, if the final system has more energy than the initial system, a mistake was made whilst solving.

Broader Impacts

The overarching goal of my proposed project is to further our collective ability to analyze computationally costly PDEs. Most <u>current methods of solving PDEs become unreasonable when the dimension of the system is large</u>. Finite difference methods, for instance, require the use of a mesh, which, in high dimensions, requires a substantial amount of memory to store, and the associated computational cost grows unreasonably large. Using a neural network in which relatively few points are needed for training, reduces the computational cost to a more reasonable size. Sirignanao and Spiliopoulos are able to solve a PDE in 200 dimensions with their approach [2]. Jiequn Han, Linfeng Zhang, and Weinan E. are able to solve the many-electron Schroedinger equation for He, H2, Be, B, LiH, and a chain of 10 hydrogen atoms [1]. However, further exploration is needed to develop a <u>more rigorous theory and understanding of convergence criteria of neural networks used to approximate solutions of PDEs</u>. Current algorithms that utilize neural networks to find solutions to PDEs also suffer from poor estimates of higher order derivatives; low variance and low bias is sacrificed for computation speed [2]. These current algorithms could be improved by use of more advanced techniques.

PDEs are used to model systems in nearly every field of research: physics, biology, sociology, finance, mathematics; however, there use is often limited, or unfulfilled due to the complexity of finding solutions to them. The use of neural networks as an aid to solving PDEs will allow for more practical application of PDEs in areas where they are currently impractical due to computational cost. Beyond using a neural network to solve PDE, the further development of theory relevant to the convergence of neural networks to a solution of PDEs will be revolutionary. My application of neural networks to solve computationally costly PDEs along with furthering knowledge of convergence criteria for neural networks will benefit the deep learning community as a whole.

References

[1] Jiequn Han, Linfeng Zhang, and Weinan E. “Solving many-Electron Schroedinger Equation Using Deep Neural Networks”. *Journal of Computational Physics (Vol. 399)*. 2019

[2] Justin Sirignano, and Konstantinos Spiliopoulos. “DGM: A deep learning algorithm for solving partial differential equations”. *Journal of Computational Physics (Vol. 375)*. 2018

[3] M. Raissi, P. Perdikaris, and G.E. Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. *Journal of Computational Physics (Vol. 378)*. 2019

[4] Leah Bar and Nir Sochen. “Unsupervised Deep Learning Algorithm for PDE-based Forward and Inverse Problems”. *ArXiv*. 2019