Unsupervised Neural Networks for Quantum Eigenvalue Problems

Henry Jin
Department of Electrical Engineering and Computer Sciences, University of California, Berkeley
Berkeley, California 94720, United States
helinjin@berkeley.edu

Marios Mattheakis, Pavlos Protopapas
John A. Paulson School of Engineering and Applied Sciences, Harvard University
Cambridge, Massachusetts 02138, United States
{mariosmat, pavlos}@seas.harvard.edu

Abstract

Eigenvalue problems are critical to several fields of science and engineering. We present a novel unsupervised neural network for discovering eigenfunctions and eigenvalues for differential eigenvalue problems with solutions that identically satisfy the boundary conditions. A scanning mechanism is embedded allowing the method to find an arbitrary number of solutions. The network optimization is data-free and depends solely on the predictions. The unsupervised method is used to solve the quantum infinite well and quantum oscillator eigenvalue problems.

1 Introduction

Differential equations are prevalent in every field of science and engineering, ranging from physics to economics. Thus, extensive research has been done on developing numerical methods for solving differential equations. With the unprecedented availability of computational power, neural networks hold promise in redefining how computational problems are solved. Among other applications, unsupervised neural networks are capable of efficiently solving differential equations [1, 3, 6, 7, 8]. These are unsupervised, data-free methods where the optimization depends solely on the network predictions. The neural network solvers pose several advantages over numerical integrators: the obtained solutions are analytical and differentiable [6], networks are more robust to the 'curse of dimensionality' [3], numerical errors are not accumulated [8], and a family of solutions corresponding to different initial or boundary conditions can be constructed [2].

Differential eigenvalue equations with boundary conditions appear in a wide range of problems, including quantum mechanics and applied mathematics. Efficient numerical iterative methods, such as finite difference method, have been developed for solving eigenvalue problems, but they share the drawbacks common to all numerical integrators. Lagaris et al. [5] have shown that neural networks are able to solve eigenvalue problems and proposed a partially iterative method that solves a differential equation with a fixed eigenvalue at each iteration. Our contribution includes a novel unsupervised neural network architecture that simultaneously learns eigenvalues and the associated eigenfunctions using a scanning mechanism. The proposed technique is an extension to neural network differential equation solvers and, consequently, acquires all the benefits that network solvers have over numerical integrators. Moreover, our method has an additional advantage over integrators in that it discovers solutions that identically satisfy the boundary conditions. We assess the performance of the proposed architecture by solving two standard eigenvalue problems of quantum mechanics, namely, the infinite well, and the quantum harmonic oscillator.

Third Workshop on Machine Learning and the Physical Sciences (NeurIPS 2020), Vancouver, Canada.

2 Methodology

We consider an eigenvalue problem that exhibits the form:

$$\mathcal{L}f(x) = \lambda f(x), \tag{1}$$

where x is the spatial variable, $\mathcal{L}$ is a differential operator which depends on x and its derivatives, $f(x)$ is the eigenfunction, and λ is the associated eigenvalue. We assume homogeneous Dirichlet boundary conditions at x_L and x_R such that $f(x_{\rm L}) = f(x_{\rm R}) = f_b$, where f_b is a constant boundary value. For a given and fixed eigenvalue, Eq. (1) is an equation that can be solved by neural network methods suggested in [1, 5, 6]. We introduce a new architecture shown in Fig. 1, which is capable of solving Eq. (1) when both $f(x)$ and λ are unknown. The network takes two inputs, x and 1. The constant input feeds a single linear neuron (affine transformation) that is updated through optimization, allowing the network to find constant λ. The x and λ feed a feed-forward fully-connected network that returns an output function $N(x, \lambda)$. The predicted eigenfunctions $f(x, \lambda)$ is defined by using a parametric trick, similar to Ref. [8], according to the equation:

$$f(x, \lambda) = f_b + g(x)N(x, \lambda), \tag{2}$$

where we employ the parametric function

$$g(x) = \left(1 - e^{-(x-x_{\rm L})}\right)\left(1 - e^{-(x-x_{\rm R})}\right), \tag{3}$$

which embeds the boundary conditions in the $f(x, \lambda)$.

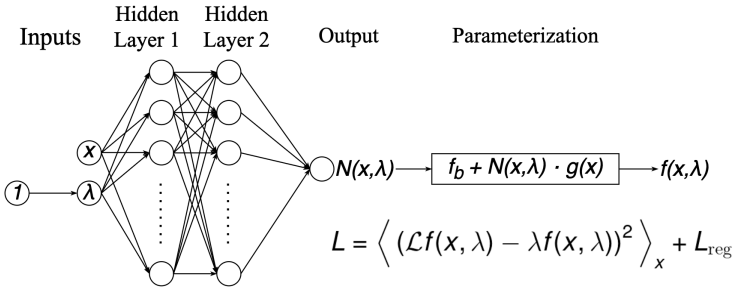

Figure 1: Adopted eigenvalue problem architecture.

Our aim is to discover pairs of $f(x, \lambda)$ and λ that satisfy Eq. (1). This is achieved by minimizing, during the optimization, a loss function L defined by Eq. (1) as:

$$\begin{aligned} L &= L_{\rm DE} + L_{\rm reg} \\ &= \left\langle \left(\mathcal{L}f(x, \lambda) - \lambda f(x, \lambda)\right)^2 \right\rangle_x + L_{\rm reg}, \end{aligned} \tag{4}$$

where $\langle\cdot\rangle_x$ represents averaging with respect to x. Any derivative with respect to x contained in $\mathcal{L}$ is calculated by using the auto-differentiation technique [9]. The $L_{\rm reg}$ in Eq. (4) contains regularization loss functions and is defined as: $L_{\rm reg} = \nu_f L_f + \nu_\lambda L_\lambda + \nu_{\rm drive} L_{\rm drive}$. Empirically, for the problems discussed below, we found the optimal set $\nu_f = \nu_\lambda = \nu_{\rm drive} = 1$. The L_f and L_λ are used to avoid learning trivial eigenfunctions and eigenvalues respectively, while $L_{\rm drive}$ motivates the network to scan for higher eigenvalues, as we explain below. The regularization functions are defined as:

$$L_f = \frac{1}{f(x, \lambda)^2}, \qquad L_\lambda = \frac{1}{\lambda^2}, \qquad L_{\rm drive} = e^{-\lambda + c}. \tag{5}$$

During the optimization, a scheduled scanning algorithm increases the $L_{\rm drive}$ by increasing c in regular intervals. That forces the network to search for larger eigenvalues and the associated eigenfunctions. At each interval the network is optimized and the model parameters are stored when sufficiently low $L_{\rm DE}$ is achieved. We emphasize that the loss function solely depends on the predictions of the network and, therefore, the training process is data-free, resulting in an unsupervised learning method. For the training, a batch of x points in the interval $[x_{\rm L}, x_{\rm R}]$ is selected as input. In every training iteration (epoch) the input points are perturbed by a Gaussian noise [8]. Adam optimizer is used [4]

with a learning rate of $8 \cdot 10^{-3}$. We use two hidden layers of 50 neurons per layer with trigonometric $\sin(\cdot)$ activation function. The use of $\sin(\cdot)$ instead of more common activation functions, such as Sigmoid$(\cdot)$ and $\tanh(\cdot)$, significantly accelerates the network's convergence to a solution [8]. We implemented the proposed neural network in pytorch [9] and published the code on github [1].

3 Experiments

We evaluate the effectiveness of the proposed method by solving the eigenvalue problem defined by Schrodinger's equation. This is a fundamental equation in quantum mechanics that describes the state wavefunction $\psi(x)$ and energy E of a quantum system. We are interested in solving the one-dimensional stationary Schrodinger equation defined as:

$$\left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x)\right]\psi(x) = E\psi(x), \tag{6}$$

where $\hbar$ and m stand for Planck constant and mass which, without loss of generality, can be set to $\hbar = m = 1$. Equation (6) defines an eigenvalue problem where $\psi(x)$ and E denote the eigenfunction $f(x, \lambda)$ and eigenvalue λ pair. A boundary value eigenvalue problem is defined by considering a potential function $V(x)$ and boundary conditions of $\psi(x)$. We assess the performance of the proposed network architecture by solving Eq. (6) for the potential functions of the infinite square well and the harmonic oscillator, both of which have known analytical solutions.

3.1 Infinite Square Well

The infinite square well problem is characterized by the following potential function:

$$V(x) = \begin{cases} 0 & 0 \leq x \leq \ell \\ \infty & \text{otherwise} \end{cases}, \tag{7}$$

where the length of the well is set to $\ell = 1$. The exact eigenfunctions and eigenvalues read

$$\psi_n(x) = \begin{cases} \sqrt{2}\sin(n\pi x) & 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}, \qquad E_n = \frac{n^2\pi^2}{2}, \tag{8}$$

where n is a positive integer and indicates different solutions. The eigenfunctions are strictly zero outside of the well, implying the boundary conditions $\psi(0) = \psi(1) = 0$. The Eqs. (2) and (3) ensure the boundary conditions by setting $x_{\mathrm{L}} = 0$, $x_{\mathrm{R}} = 1$, and $f_b = 0$. The proposed scanning model is capable of solving for an arbitrary number of the first n states. In Fig. 2 we show results up to $n = 3$. The left panel presents the loss functions of Eqs. (4) and (5) (upper), and the predicted E (lower) during the network optimization. The scanning algorithm pushes the predicted eigenvalue upwards. The loss falls precipitously when an eigenfunction is found, and the energy shows plateaus at these exact eigenvalues (indicated by dashed black line) of Eq. (8). The loss function in Fig. 2 depicts three dips, which correspond to three plateaus in the energy. This behavior gives a physical meaning to the loss function, since by inspecting L during the training we can draw the eigenstates. The right panel shows the extracted $\psi(x)$ (blue) and E (dashed black) at each plateau. Comparing with the exact solutions of Eq. (8), the order of magnitude of errors are 10^{-3} and 10^{-4} for ψ_n and E_n, respectively.

3.2 Quantum Harmonic Oscillator

The harmonic oscillator is characterized by the quadratic potential function

$$V(x) = \frac{1}{2}kx^2, \tag{9}$$

where k is the force constant and is considered to be $k = 4$. The exact solutions for the eigenfunctions and energies are given in terms of Hermite polynomials H_n as

$$\psi_n(x) = \frac{1}{\sqrt{2^n n!}}\frac{e^{-\frac{x^2}{2}}}{\pi^{1/4}}H_n(x), \qquad E_n = n + \frac{1}{2}. \tag{10}$$

[1] https://github.com/henry1jin/eigeNN

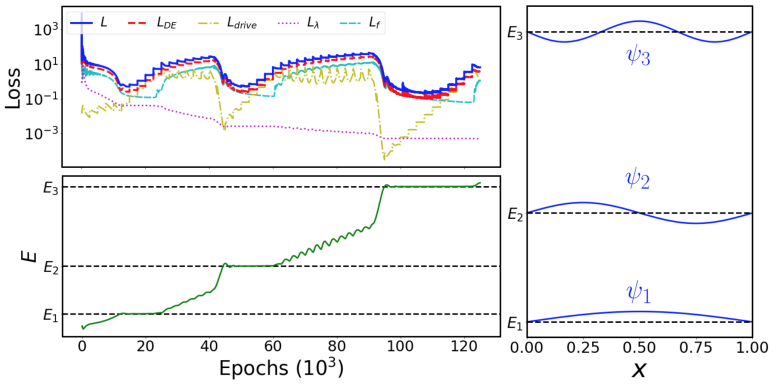

Figure 2: Infinite square well: Left panel shows the loss functions and the predicted energy during the training; dashed lines indicate the exact energy levels. Right plot outlines the predicted eigenfunctions (blue) and eigenvalues (dashed black). The errors are of the order 10^{-3} and 10^{-4} for ψ and E.

The boundary conditions for the quantum oscillator problem dictate the wavefunction to vanish at infinity, that is, $\psi(-\infty) = \psi(\infty) = 0$. In numerical methods, infinity is assumed to be a large number compared to the potential dimensions. We adopt the same approach and consider the boundary conditions $\psi(-6) = \psi(6) = 0$. Thus, Eqs. (2) and (3) ensure the boundary conditions by setting $x_{\mathrm{L}} = -6$, $x_{\mathrm{R}} = 6$, and $f_b = 0$.

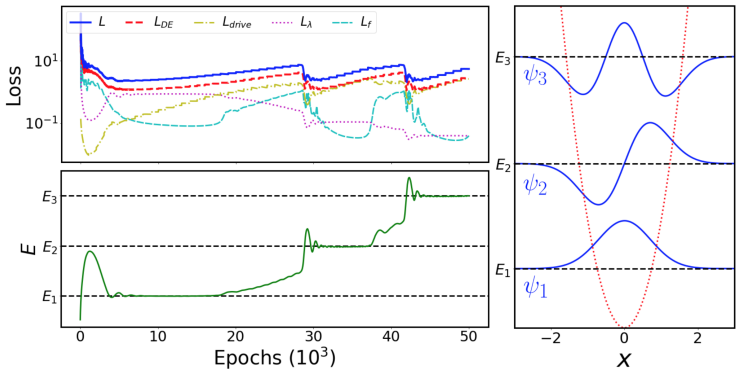

Figure 3: Quantum harmonic oscillator: Top left shows the various loss terms. Bottom left plots the history of the predicted energy. Right plot shows the eigenfunctions found by the model; the red dotted line outlines the potential. The order of magnitude of errors are 10^{-2} for ψ and 10^{-2} for E.

The proposed scanning neural network method is employed to discover the first three eigenstates for the quantum harmonic oscillator. The left panel in Fig. 3 shows the drops in the total loss (blue line) that correspond to plateaus in the eigenvalue during training (green), indicating that an eigenvalue has been found. The predicted $\psi_n(x)$ and E_n are presented in the right panel in Fig. 3 by solid blue and dashed black lines, respectively, while the red dotted curve outlines the potential energy.

4 Conclusion

In recent years, there has been a growing interest in the application of neural networks to study differential equations. In this work, we introduced a neural network that is capable of discovering eigenvalues and eigenfunctions for boundary conditioned differential eigenvalue problems. The obtained solutions identically satisfy the given boundary conditions. A scanning mechanism allows the network to find an arbitrary number of eigenvalues and associated eigenfunctions. Inspecting the loss function during training allows one to draw the eigenstates, providing a physical meaning to the loss function. The optimization solely depends on the network's predictions, consisting of an unsupervised learning method. We demonstrated the capability of the proposed architecture by solving the infinite well and harmonic oscillator quantum problems.

This work presents a parametrization that allows the network to discover solutions that identically satisfy the homogeneous Dirichlet boundary conditions. Using alternative parametric forms yields network predictions that ensure other type of constraints such as non-homogeneous Dirichlet, Neumann, and mixed boundary conditions. The embedded scanning mechanism is characterized by a set of hyperparameters that determine the robustness of the network in discovering different eigenvalues-eigenfunctions pairs. In the future, we will extend the scanning method to adjust the hyperparameters with respect to the problem. This optimization will make the training more efficient and will make the network more robust in the discovery of eigenvalues and eigenfunctions.

Broader Impact

This work is valuable for computational physicists and applied mathematicians, as well as, in any field where differential eigenvalue problems may arise. We have demonstrated our method's success for the one dimensional Schrodinger equation, but the technique can be generalised to Sturm-Liouville problems, as well as higher dimensional equations (e.g. 3D Schrodinger and Helmholtz equations). We strongly believe that this study will serve as the groundwork for future works in the area of solving differential equations using deep learning methods. We neither foresee and nor desire our research results to be used for any kind of discrimination.

Acknowledgments and Disclosure of Funding

The authors would like to acknowledge fruitful discussions with Dr. David Sondak. We also thank the anonymous reviewers for their valuable comments on this manuscript.

References

[1] Feiyu Chen, David Sondak, Pavlos Protopapas, Marios Mattheakis, Shuheng Liu, Devansh Agarwal, and Marco Di Giovanni. Neurodiffeq: A python package for solving differential equations with neural networks. *Journal of Open Source Software*, 5(46):1931, 2020.

[2] Cedric Flamant, Pavlos Protopapas, and David Sondak. Solving differential equations using neural network solution bundles. volume 2006.14372, 2020.

[3] Jiequn Han, Arnulf Jentzen, and E Weinan. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences of the United States of America*, 115 34:8505–8510, 2017.

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[5] I.E. Lagaris, A. Likas, and D.I. Fotiadis. Artificial neural network methods in quantum mechanics. *Computer Physics Communications*, 104(1):1 – 14, 1997.

[6] Isaac Lagaris, Aristidis Likas, and Dimitrios Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *Neural Networks, IEEE Transactions on*, pages 987 – 1000, 10 1998.

[7] Martin Magill, Faisal Qureshi, and Hendrick W. de Haan. Neural networks trained to solve differential equations learn general representations. In *NeurIPS*, 2018.

[8] Marios Mattheakis, David Sondak, Akshunna S. Dogra, and Pavlos Protopapas. Hamiltonian neural networks for solving differential equations, 2020.

[9] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary Devito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.