ASSIGNMENT IV
Physics-informed Neural Networks

SH2150 Machine Learning in Physics

Mats Persson and Joel Wållberg

December 2025

In this lab exercise, we will explore two types of problems that are attracting a lot of research interest in scientific machine learning: solving differential equations, and inverse problems.

We will study a problem in nanomaterials design. Let us study a one-dimensional "quantum dot", a material where a single electron is free to move in a one-dimensional potential. Let us assume that we are able to control the atomic composition of the quantum dot in detail such that we are able to engineer this potential to have any shape that we want. The electron cannot leave the quantum dot, so we can model the system as a one-dimensional potential well with an uneven bottom. The motion of the electron is governed by the Schrödinger equation

$$-\frac{\hbar^2}{2m}\frac{\partial^2\psi(x)}{\partial x^2} + V(x)\psi(x) = E\psi(x) \tag{1}$$

where $\hbar = h/2\pi$ (h = Planck's constant), m is the mass of the electron, $\psi(x)$ is the wavefunction and E is the energy level.

We are now interested in engineering the energy levels and the wavefunctions of this electron by constructing an appropriate potential. In order to do this, we first need to be able to find the energy levels and wavefunctions for a given potential.

You are provided a training set of potential energy functions $V(x)$ on an interval of length $L = 5$ nm, -2.5 nm $< x < 2.5$ nm, discretized with interval $\Delta x = 0.10$ nm. For $x < -2.5$ nm and $x > 2.5$ nm the potential is assumed to be infinite. Together with these potentials with associated energy levels (sorted in ascending order) and the corresponding wavefunctions, normalized such that $\int |\psi(x)|^2 \mathrm{d}x = 1$. Note that, even though quantum mechanical wave functions are generally complex-valued, we are only concerned with time-independent solutions in this case and therefore we can without loss of generality assume that all wavefunctions are real-valued.

Apart from this training set, you are also provided a validation set of potentials and energy levels, as well as an "out-of-distribution" test set with 10 potentials that are different from the potentials in the training set, together with the corresponding energy levels and wave functions.

You are also provided noisy versions of the above datasets, to check the robustness to noise. (The noise in the potentials, energies and eigenvectors is independent.)

Your tasks are to

- Study different approaches to solving the Schrödinger equation using machine learning

- Investigate whether the combination of training data and physics information gives a better solution than either of these individually.
- Investigate if it is possible to use machine learning to solve the *inverse* problem of reconstructing the potential from the energy levels, and if having physics information is helpful in this regard.

Problems 3-5 below have a more open-ended nature. There may not always be a single "correct way" to do it but rather you will be expected to explore the problem and report what you find.

1. We will start by investigating if it's possible to learn a direct mapping from the potential function to the energy levels. For this, we will make use of convolutional neural networks.

   (a) Use the provided set of potentials and corresponding energy levels to train a CNN to map the potential to the set of energy levels. Apply this to the first validation potential. How accurate is the solution? (2p)

   (b) The inference time alone is relevant in situations where the time from data availability to solution is critical, for example if the differential equation solver is part of a real-time feedback loop. Measure the training time and inference time. Also measure how long time it takes to find the energy levels analytically using the function `numpy.linalg.eig`. What is the speedup factor by using the learned model? (2p)

   (c) In other applications, it is more relevant to look at the total time for training and inference. How many inferences do you need to make before the time savings by using the learned mapping outweighs the time required for training? How does the answer change if you also include the time to generate the solutions in the training? (1p)

   (d) The provided dataset also contains a set of out-of-distribution potentials. How are these different from the potentials in the training dataset? Is the model performance similar for the out-of-distribution potentials as for those of the training set, or is there a difference? (1p)

   (e) Now let us turn to the wave function. Can you also train a neural network for predicting the ground state of the system, i.e. the wave function corresponding to the lowest energy level, together with the corresponding eigenvalue? (2p)

2. Before we take the next step, let us explore the concept of incorporating physics information by solving a simple "toy" version of this problem to explore the concept of physics-informed machine learning in a setting where it is easy to analyze what is happening.

   (a) Let us discretize the Schrödinger equation with only two points x_1 and x_2 and let $\psi_1 = \psi(x_1)$, $\psi_2 = \psi(x_2)$ $V_1 = V(x_1)$ and $V_2 = V(x_2)$. It will be useful to also define $V_0 = \frac{1}{2}(V_1 + V_2)$ and $\Delta V = V_2 - V_1$. Show that this gives a discretized Schrödinger equation of the form (1p)

$$\begin{bmatrix} -2d + V_1 & d \\ d & -2d + V_2 \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix} = E \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix} \tag{2}$$

   What is the value of d? We will denote $H = \begin{bmatrix} -2d + V_1 & d \\ d & -2d + V_2 \end{bmatrix}$.

   (b) Verify that the energy levels (eigenvalues) of 2 are (1p)

$$E = -2d + V_0 \pm \sqrt{d^2 + \left(\frac{\Delta V}{2}\right)} \tag{3}$$

by showing that they satisfy the characteristic equation $\det(EI - H) = 0$. (det denotes determinant and I is the identity matrix.) Also verify that the wavefunction (eigenvector) $[\psi_1 \ \psi_2]^T$ corresponding to eigenvalue E satisfies

$$\psi_2/\psi_1 = 2 + \frac{E - V_1}{d} \tag{4}$$

and derive expressions for ψ_1 and ψ_2 using the normalization condition that $\psi_1^2 + \psi_2^2 = 1$ (note that the answer is not unique).

(c) Now that we have the analytical solution to this toy problem, we want to see if a neural network can learn to predict the energy levels and wave functions from the parameters V_0 and ΔV. Create a training set by using the analytical solution above to calculate the energies and wavefunctions corresponding to the three data points $(V_0, \Delta V) = (0, d/2), (-d, d), (0, 2d)$. Then train a small multilayer perceptron with two inputs V_0 and ΔV to predict three outputs: E, ψ_1 and ψ_2. (You can start trying with a network with 20 neurons in total, using MSE loss, and modify as needed.) Make three plots with the resulting predictions $of E, \psi_1, \psi_2$ as a function of ΔV for $V_0 = 0$ and compare to the analytical result. (2p)

(d) Now modify the above network by including physics information. To do this, add two additional terms to the loss function $||H - \lambda E||^2$ and $(\psi_1^2 + \psi_2^2 - 1)^2$. You will need to add more test points $(V_0, \Delta V)$ where these terms will be evaluated. (2p)

(e) What happens to the networks with and without physics information if you include a small model error? Add a constant term to V_0 for all training points, to model a measurement that does not quite agree with the model, and see if the combination of training points and physics information can combine the robustness of physics-informed learning with a fine-tuning to measured data. (1p)

3. Now let us try to modify the model we developed for estimating the energy and wavefunction in question 1 by incorporating physics information in the network.

   (a) Add terms proportional to $||H\psi - E\psi||^2$ and $(||\psi||^2 - 1)^2$ to the network that you trained for estimating the ground state. Compare to the solution from problem 1. (2p)

   (b) Now use the "noisy potentials" dataset to simulate a situation where the potential is not perfectly known but has a measurement error. Compare the accuracy you achieve with the unrolled iterative method and the one from problem 1 for this set of potentials. Which is more robust to noise in the input dataset? (1p)

   (c) Use subsets of different size, e.g., $(N = 10, 20, 50, 100)$ of the "noisy potentials" dataset to retrain these two types of CNNs, and compare the performance on the validation set as function of N. (2p)

4. Another approach to solving this problem with machine learning is by using a physics-informed neural network (PINN).

   (a) Write down the loss function of a neural network that solves the Schrödinger equation for this problem including boundary conditions ($\psi(x) = 0$ at the endpoints). You will have to assume that E is a fixed parameter. (1p)

   (b) Train a neural network that minimizes this loss function, for the first potential in the validation dataset. Repeat this for a range of energies E and plot the loss as a function of E. Can you use this plot to identify the three lowest energy levels of the system? Hint: you may want to use the fact that you already know the true energy levels to restrict the search to the right energy ranges. (Are the eigenvalues given by the PINN guaranteed to be identical to the ones obtained by the analytical solution?) (2p)

5. Now that we have explored different ways of solving the Schrödinger equation with machine learning, we turn to the *inverse* problem: given a set of energy levels, how can we construct the potential that will give this set of energy levels?

   (a) Use the set of potentials and solutions to train a network that maps an ordered vector of eigenvalues to a potential energy function. Calculate the MSE loss on the validation set and plot the potential recovered from the first set of energy levels in the validation set. (2p)

   Final question: Did you use an AI tool (other than the machine learning models you trained in this exercise) for anything else than information searching, when solving this problem set? If so, please write a **brief statement of how you used AI.**

Total number of points: 25

Motivate your answers wherever applicable.

Good luck!