Physics Informed Neural Networks

Mehdi Yaghouti

November 4, 2023

1 Introduction

Neural networks have proven as powerful tools in leveraging data to solve a wide variety of complex problems. Their ability to uncover patterns by processing vast amounts of data makes them invaluable in various engineering domains. However, despite of their exceptional performance in several domains, their substantial data requirements impose severe constraints on their utility in numerous critical applications. As a significant instance, real world dynamical systems are often involve nonlinear behaviors governed by physical laws and differential equations. Conventional neural networks typically relies on availability of substantial amount of collected data to approximate the behavior of such systems, which can be expensive or even unattainable in many real-world scenarios.

Physics-informed neural networks (PINNs) revolve around the central idea of integrating domain-specific knowledge and physical laws into their architecture to offer a distinct advantage over traditional neural networks in solving differential equations. By incorporating the mathematical structure of equations and constraints, PINNs can solve engineering problems only using limited data, making them invaluable when data is scarce or uncertain. In simple word they combine the data-driven skill of Neural Networks with the domain knowledge of the problem at hand to provide more accurate and dependable predictions in solving dynamical systems.

Below we try to showcase and justify the advantage of using PINNs over conventional Numerical Solvers and purely data driven Neural Networks by comparing the accuracy and required computational intensity between the alternative approaches applied on two simple toy model dynamical systems.

2 Toy Model I: Damped Harmonic Oscillator

As our first toy model, consider the mass-spring-damper system depicted in Figure 1. Using Newton's law the dynamics of the system can be formulated as follows:

$$m\,\ddot{x}(t) = -k\,x(t) - c\,\dot{x}(t) \tag{1}$$

Introducing the parameters $\delta = \frac{c}{2\,m}$ and $\omega = \sqrt{{\omega_0}^2 - \delta^2}$, the exact solution to the above equation in underdamped regime $\delta < \omega_0$ is given by,

$$x(t) = 2\,A\,e^{-\delta\,t} cos\,(\phi + \omega\,t) \tag{2}$$

where the oscillation amplitude A and phase ϕ can be calculated from the initial conditions as follows,

$$\begin{cases} x(0) & = 2\,A\,cos(\phi) \\ \frac{d\mathbf{x}(0)}{dt} & = -2\,A\,(\delta\,cos(\phi) + \omega\,sin(\phi)) \end{cases}$$

However, unlike this simple differential equation at hand, many differential equations do not have closed-form solutions, necessitating the use of numerical approximation methods to solved them in practice. Usually it is benificial to recast the equation to the equivalent first order dynamical system of the following form before applying any numerical solver,

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}(t), t), \qquad \text{with } \mathbf{x}(t_0) = \mathbf{x}_0 \tag{3}$$

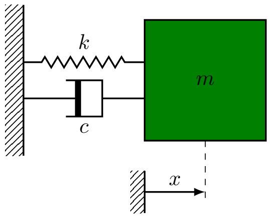

Figure 1: Damped Harmonic Oscillator, $m\ddot{x}(t) = -kx(t) - c\dot{x}(t)$

A simple implementation of the well-known 4th-order Runge-Kutta solver, requires calling the function $\mathbf{f}$ four times to calculate the quantities $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3$ and $\mathbf{t}_4$ as follows,

$$\begin{cases} \mathbf{t}_1 & = \mathbf{f}(\mathbf{x}(t_0), t_0) \\ \mathbf{t}_2 & = \mathbf{f}\left(\mathbf{x}(t_0) + \mathbf{t}_1\frac{h}{2}, t_0 + \frac{h}{2}\right) \\ \mathbf{t}_3 & = \mathbf{f}\left(\mathbf{x}(t_0) + \mathbf{t}_2\frac{h}{2}, t_0 + \frac{h}{2}\right) \\ \mathbf{t}_4 & = \mathbf{f}\left(\mathbf{x}(t_0) + \mathbf{t}_3 h, t_0 + h\right) \\ \mathbf{x}(t_0 + h) & = \mathbf{x}(t_0) + \frac{h}{3}\left(\frac{1}{2}\mathbf{t}_1 + \mathbf{t}_2 + \mathbf{t}_3 + \frac{1}{2}\mathbf{t}_4\right) \end{cases}$$

In the current case of a linear ODE, each function call is equivalent to a matrix-vector multiplication of the order of the equation. Therefore, in the case of the second order ODE at hand the above calculation involves 22 scalar multiplications + 15 scalar additions for each time steps. Figure **??** plotted the displacement x as a function of time resulted by solving the given differential equation using 4th-order Runge-Kutta method.

As an alternative to traditional numerical solvers, the capability of Neural Networks as powerful function approximators can be exploited to learn the solution of a differential equation. However, neural networks typically rely on availability of significant amount of data samples to approximate the solution of such equations. Figure 3, demonstrates this limitation by plotting the output of a four layers deep Neural Network trained on only a few initial samples of damped harmonic oscillator solution. The structure of the neural network is depicted in Figure 2. The network is trained on only 20 data samples using the mean square loss function,

$$\mathscr{L} = \frac{1}{N_t}\sum_{j=1}^{N_t} \mid x(t_j) - \hat{x}(t_j) \mid^2 \tag{4}$$

Figure 3 clearly shows that the network output begins to diverge from the correct solution beyond the domain covered by training samples.

To highlight the effectiveness of physics-informed neural networks, Figure 3 also illustrates the output of the identical neural network not only trained on the same data samples but also by imposing a penalty on the equation structure,

$$\mathscr{L} = \frac{1}{N_s}\sum_{j=1}^{N_s} \mid x(t_j) - \hat{x}(t_j) \mid^2 + \frac{1}{N_e}\sum_{j=1}^{N_e} \mid \mathscr{F}\left(x(t_j)\right) \mid^2 \tag{5}$$

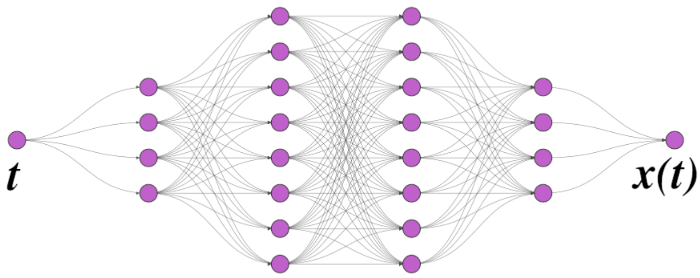

Figure 2: Four layers deep fully connected Neural Network

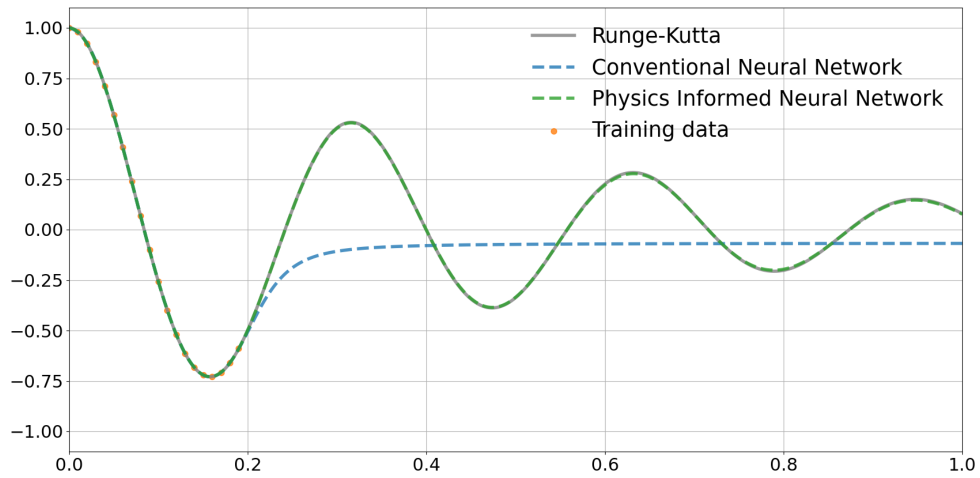

Figure 3: The comparison between the results of conventional Neural Network and Phyiscs Informed Neural Network for damped harmonic oscillator

where we have introduced $\mathscr{F}\left(x(t_j)\right) := m\frac{d^2x(t_j)}{dt^2} + c\,\frac{dx(t_j)}{dt} + k\,x(t_j)$ to impose a penalty on any deviation from the underlying mathematical structure of the equation. As the results show, the physics informed network show a perfect match with the correct solution well beyond the domain covered by the training samples.

3 Toy Model II: Coupled Double mass-Spring-Damper System

For the second example, we consider the coupled double mass-spring-damper system depicted in Figure 1. Using Newton's law the dynamics of this system can be formulated as follows:

$$\begin{cases} m_1\ddot{x}_1 & = -k_1x_1 - c_1\dot{x}_1 + k_2\left(x_2 - x_1\right) + c_2(\dot{x}_2 - \dot{x}_1) \\ m_2\ddot{x}_2 & = -k_2\left(x_2 - x_1\right) - c_2\left(\dot{x}_2 - \dot{x}_1\right) \end{cases}$$

The two ODEs are each associated with one degree of freedom x_1 and x_2 in the physical system. It is evident that in this case we need a neural network with dual outputs to model a time-dependent vector function with two degrees of freedom. Figure 5 depicts a schematic of a four layers deep neural network with two outputs $x_1(t)$ and $x_2(t)$. Similar to the damped harmonic oscillator example, the loss function can be expressed as the summation of multiple terms. The first and second terms address the initial boundary conditions, while each of the remaining two terms enforces the mathematical structure of one of the ODEs.

$$\mathscr{L} = \frac{1}{N_1}\sum_{j=1}^{N_1}||\mathbf{x}(0) - \hat{\mathbf{x}}(0)||^2 + \frac{1}{N_2}\sum_{j=1}^{N_2}||\frac{d\mathbf{x}}{dt}(0) - \dot{\mathbf{x}}(0)||^2 + \frac{1}{N_3}\sum_{j=1}^{N_3}|\ \mathscr{F}_1\left(\mathbf{x}(t_j)\right)|^2 + \frac{1}{N_4}\sum_{j=1}^{N_4}|\ \mathscr{F}_2\left(\mathbf{x}(t_j)\right)|^2 \tag{6}$$

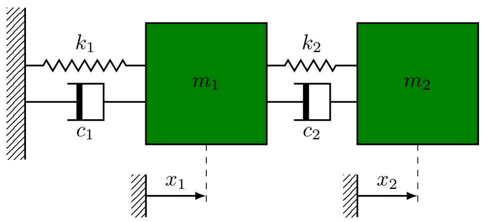

Figure 4: Coupled Double mass-spring-damper system

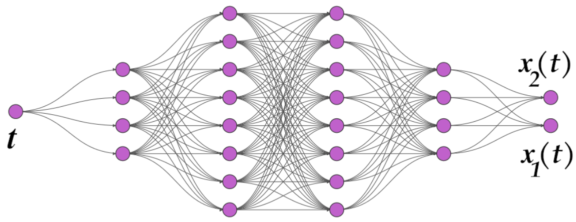

Figure 5: The schematic of the neural network with two outputs

where,

$$\begin{cases} \mathscr{F}_1\left(\mathbf{x}(t_j)\right) & = m_1 \frac{d^2 x_1(t_j)}{dt^2} + k_1\, x_1(t_j) + c_1 \frac{dx_1(t_j)}{dt} - k_2\left(x_2(t_j) - x_1(t_j)\right) - c_2\left(\frac{dx_2(t_j)}{dt} - \frac{dx_1(t_j)}{dt}\right) \\ \mathscr{F}_2\left(\mathbf{x}(t_j)\right) & = m_2 \frac{d^2 x_2(t_j)}{dt^2} + k_2\left(x_2(t_j) - x_1(t_j)\right) + c_2\left(\frac{dx_2}{dt}(t_j) - \frac{dx_1}{dt}(t_j)\right) \end{cases}$$

Figure 6 presents a comparison between the results obtained using a physics-informed neural network and those obtained by training a neural network only on the training data, without enforcing any mathematical constraints.

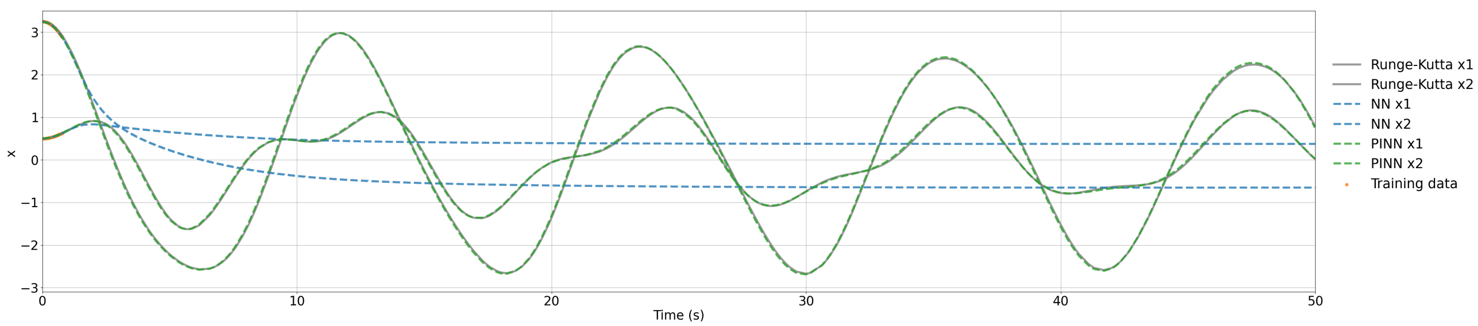

Figure 6: The comparison between the results of conventional Neural Network and Phyiscs Informed Neural Network for double mass damper system with parameters $m_1 = 6.4$, $m_2 = 4.4$, $k_1 = 5.0$, $k_2 = 2.0$, $c_1 = 0.1$, $c_2 = 0.08$