Physics Informed Deep Learning

Yikai Wang

September 28, 2021

1 Problem Setup

Consider parametrized and nonlinear partial differential equations of the general form

$$u_t + \mathcal{N}[u;\lambda] = 0$$

where $u(t,x)$ denotes the latent(hidden) solution and $\mathcal{N}[\cdot;\lambda]$ is a nonlinear operator parametrized by λ.

1.1 Continuous Time Models

Define $f(t,x)$ to be given by the left-hand-side of equation; ie.,

$$f := u_t + \mathcal{N}[u], \tag{1}$$

and proceed by approximating $u(t,x)$ by a deep neural network.

1.1.1 **Example**(*Burger's Equation*)

$$\begin{aligned} &u_t + uu_x - (0.01)/\pi)u_{xx} = 0, x \in [-1,1], t \in [0,1] \\ &u(0,x) = -sin(\pi x), \\ &u(t,-1) = u(t,1) = 0. \end{aligned} \tag{2}$$

Let us define $f(t,x)$ to be given by

$$f := u_t + uu_x - (0.01/\pi)u_{xx} \tag{3}$$

The shared parameters between the neural networks $u(t,x)$ and $f(t,x)$ can be learned by minimizing the mean square error loss

$$MSE = MSE_u + MSE_f \tag{4}$$

where

$$MSE_u = \frac{1}{N_u}\sum_{i=1}^{N_u} |u(t_u^i, x_u^i) - u^i|^2$$

and

$$MSE_f = \frac{1}{N_f}\sum_{i=1}^{N_f} |f(t_f^i, x_f^i|^2$$

Here $\{t_u^i, x_u^i, u^i\}_{i=1}^{N_u}$ denote the initial and boundary training data on $u(t,x)$ and $\{t_f^i, x_f^i\}_{i=1}^{N_f}$ specify the collections points for $f(t,x)$.

1.1.2 **Example**(*Shrödinger Equation*)

The nonlinear Schrödinger equation along with periodic boundary conditions is given by

$$\begin{aligned} &ih_t + 0.5h_{xx} + |h|^2 h = 0, x \in [-5,5], t \in [0, \pi/2] \\ &h(0,x) = 2sech(x), \\ &h(t,-5) = h(t,5), \\ &h_x(t,-5) = h_x(t,5), \end{aligned} \tag{5}$$

Where $h(t,x)$ is the complex-valued solution. Define $f(t,x)$ to be given by

$$f := ih_t + 0.6h_{xx} + |h|^2 h$$

and proceed by placing a complex-valued nerual network prior on $h(t,x)$. In fact, if u denotes the real part of h and v is the imaginary part, we are placing a muti-out neural network prior on $h(t,x) = [u(t,x)\ v(t,x)]$. The shared parameters of the neural networks $h(t,x)$ and $f(t,x)$ can be learned by minimizing the mean square error loss

$$MSE = MSE_0 + MSE_b + MSE_f \tag{6}$$

where

$$MSE_0 = \frac{1}{N_0}\sum_{i=1}^{N_0}|h(0,x_0^i) - h_0^i|^2$$

$$MSE_b = \frac{1}{N_b}\sum_{i=1}^{N_b}|h^i(t_b^i,-5) - h^i(t_b^i,5)|^2 + |h_x^i(t_b^i,-5) - h_x^i(t_b^i,5)|^2)$$

, and

$$MSE_f = \frac{1}{N_f}\sum_{i=1}^{N_f}|f(t_f^i,x_f^i)|^2$$

Here $\{x_0^i, h_0^i\}_{i=0}^{N_0}$ denotes the initial data, $\{t_b^i\}_{i=1}^{N_b}$ corresponds to the collocation points on the boundary, $\{t_f^i, x_f^i\}$ represents the collocation points on $f(t,x)$. Consequently, MSE_0 corresponds to the loss on the initial data, MSE_b enforces the periodic boundary conditions, and MSE_f penalizes the Schrödinger equation not being satisfied on the collocation points.

1.2 Discrete Time Models

Apply the general form of Runge-Kutta methods with q stages to equation(1) and obtain

$$\begin{aligned} u^{n+c_i} &= u^n - \Delta t\sum_{j=1}^q a_{ij}\mathcal{N}[u^{n+c_j}], i = 1,\dots,q, \\ u^{n+1} &= u^n - \Delta t\sum_{j=1}^q b_j\mathcal{N}[u^{n+c_j}]. \end{aligned} \tag{7}$$

Here, $u^{n+c_j}(x) = u(t^n + c_j\Delta t, x)$ for $j = 1,\dots,q$. Equations (7) can be equivalently expressed as

$$\begin{aligned} u^n &= u_i^n, i = 1,\dots,q, \\ u^n &= u_{q+1}^n, \end{aligned} \tag{8}$$

where

$$\begin{aligned} u_i^n &:= u^{n+c_i} + \Delta t\sum_{j=1}^q a_{ij}\mathcal{N}[u^{n+c_j}], i = 1,\dots,q, \\ u_{q+1}^n &:= u^{n+1} + \Delta t\sum_{j=1}^q b_j\mathcal{N}[u^{n+c_j}] \end{aligned} \tag{9}$$

Then they proceed by placing a muti-output neural network prior on

$$[u^{n+c_1}(x),\dots,u^{n+c_q}(x),u^{n+1}(x)]. \tag{10}$$

This prior assumption along with equations (9) result in a neural network that takes x as input and outputs

$$[u_1^n(x),\dots,u_q^n(x),u_{q+1}^n(x)]. \tag{11}$$

1.2.1 **Example**(*Burgers' Equation*)

The nonlinear operator in equation (9) is given by

$$\mathcal{N}[u^{n+c_j}] = u^{n+c_j}u_x^{n+c_j} - (0.01/\pi)u_{xx}^{n+c_j}$$

The shared parameters of the neural networks (10) and (11) can be learned by miniming the sum of squared errors

$$SSE = SSE_n + SSE_b \tag{12}$$

where

$$SSE_n = \sum_{j=1}^q\sum_{i=1}^{N_n}|u_j^n(x^{n,i}) - u^{n,i}|^2$$

and

$$SSE_b = \sum_{i=1}^q(|u^{n+c_i}(-1)|^2 + |u^{n+c_i}(1)|^2) + |u^{n+1}(-1)|^2 + |u^{n+1}(1)|^2$$

. Here, $\{x^{n,i}, u^{n,i}\}_{i=1}^{N_n}$ corresponds to the data at time t^n.

2 Question

1. How to divide train dataset and test dataset?