Respecting Regularity Is What You Need for Training PINN

Chuwei Wang

School of Mathematical Sciences

Peking University

2022.5.10

Framework of PINN

Physics Informed Neural Network(PINN) is a widely-used method for solving differential equations (PDE and ODE).

Its framework could be generalized as below:

$$\begin{cases} Lu = f, & x \in \Omega \\ Bu = g, & x \in \partial\Omega \end{cases}$$

Minimize $loss(:= \|Lu_\theta - f\|_X + \lambda \|Bu_\theta - g\|_Y)$ w.r.t θ to solve the equation.

- θ: NN parameters
- $\|\cdot\|_X, \|\cdot\|_Y$: two norms. In PINN, L^2 norm is chosen.

Error Analysis

In practice, $\|\cdot\|_X$ is computed with quadrature or Monte Carlo (i.e. with finite sampling).

The total error of PINN could be divided into three parts.

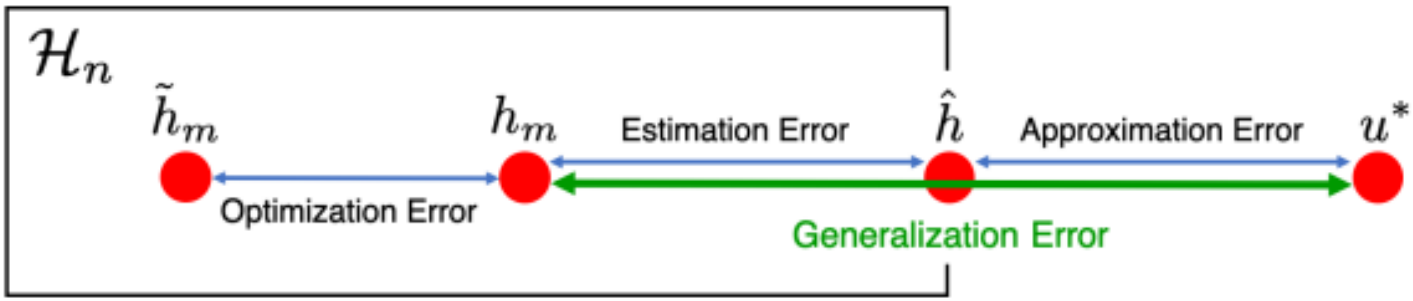
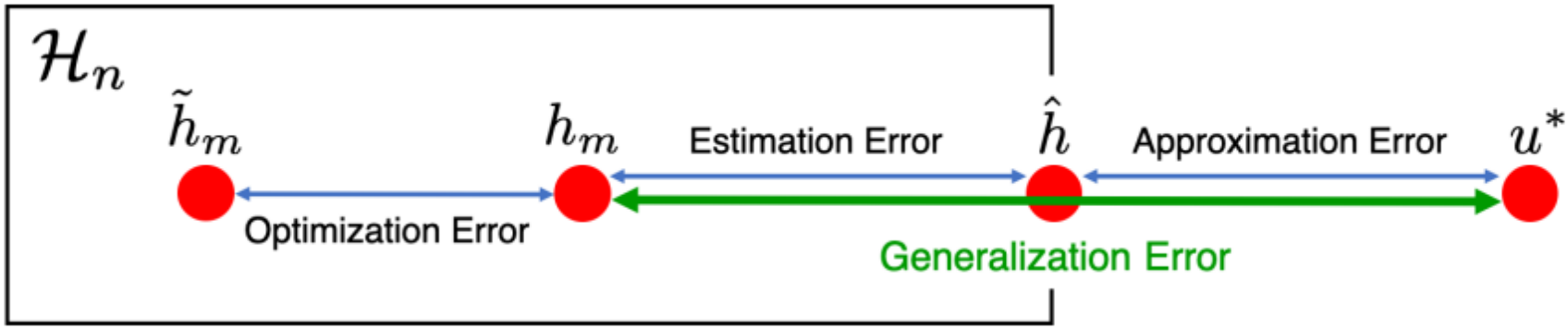


Figure 2: Illustration of the total errors. $\mathcal{H}_n$ is the chosen function class. u^* is the solution to the underlying PDE. The number of training data is m. h_m is a minimizer of the loss with m data. $\hat{h}$ is a function in $\mathcal{H}_n$ that minimizes the loss with infinitely many data. $\tilde{h}_m$ is an approximation that one obtains in practice, e.g., the result obtained after 1M epochs of a gradient-based optimization.

Error Analysis



(i) Optimization Difficulties: derivatives might cause complicated landscape.

(ii) Estimation Error: the clearest part in the three errors.

(iii) The expressive power of NN.

Stability problem:

Does there exist three norms $\|\cdot\|_X, \|\cdot\|_Y$, *and* $\|\cdot\|_Z$ *such that small* $\|Lu_\theta - f\|_X$ *and* $\|Bu_\theta - g\|_Y$ *guarantees that* $\|u_\theta - u_0\|_Z$ *is small?*

Concepts about regularity

Definition (Sobolev Norm and Sobolev Space)

- $W^{m,p}(\Omega)(m \in \mathbb{N},\ p \in [1,\infty], \Omega \subset \mathbb{R}^n)$:

$$\{f(x) \in L^p(\Omega) : D^\alpha f \in L^p(\Omega),\ \forall \alpha \in \mathbb{N}^n \text{ with } |\alpha| \leq m\}. \tag{1}$$

- $\|f\|_{W^{m,p}(\Omega)} := (\sum_{|\alpha|\leq m} \|D^\alpha f\|^p_{L^p(\Omega)})^{\frac{1}{p}}$ for $1 \leq p < \infty$, and $\|f\|_{W^{m,\infty}(\Omega)} := \max_{|\alpha|\leq m} \|D^\alpha f\|_{L^\infty(\Omega)}$
- $W_0^{m,p}(\Omega)$: the completion of $C_0^\infty(\Omega)$ under $\|\cdot\|_{m,p}$ norm.
- $H^m(\Omega) := W^{m,2}(\Omega)$, $H_0^m(\Omega) := W_0^{m,2}(\Omega)$

Informally, we could simply understand $W_0^{m,p}$ functions as $W^{m,p}$ functions with compact support.

Concepts about regularity

Definition (Sobolev Norm and Sobolev Space)

- $W^{-m,q}(\Omega)(m \in \mathbb{N},\ p \in [1,\infty],\ \frac{1}{p}+\frac{1}{q}=1,\ \Omega \subset \mathbb{R}^n)$: the dual space of $W_0^{m,p}(\Omega)$
- $||f||_{H^s} := (\int_{\mathbb{R}^n}(1+|\xi|^2)^s|\mathscr{F}f(\xi)|^2 d\xi)^{\frac{1}{2}}$

Example (Understanding $W^{-m,q}$)

Suppose $u = \nabla F$, then for any $v \in W_0^{1,p}(\Omega)$,

$|\int_\Omega uv dx| = |-\int_\Omega F \cdot \nabla v dx| \leq C(n)||v||_{1,p}||F||_q$, thus $\|u\|_{-1,q} \leq C(n)||F||_q$.

- Two definitions for H^s is equivalent when $s \in \mathbb{N}$.
- Indices p, q, m, s represent the regularity of a function.

Two basic questions regarding stability problem

(i) Could minimizing loss function with low regularity work?

(ii) Would there be advantages to choose loss functions with high regularity?(e.g., higher accuracy, less time for training, etc)

Higher precision

Theorem

Suppose Ω is bounded. Let u be the solution to

$$\begin{cases} \partial_t u - Au = f(x,t), & (x,t) \in Q_T \\ u = 0, & (x,t) \in \partial_p Q_T, \end{cases}$$

where A is a second order elliptic operator with constant parameters. Then for $2 \leq p < \infty$, if $f \in L^p(Q_T)$, we have $u \in W^{2,p}(Q_T)$ and there exists a constant C such that $\|u\|_{2,p} \leq C\|f\|_p$. ($Q_t$: $\Omega \times [0,t]$. $\partial_p Q_t$: $\Omega \times \{0\} \cup \partial\Omega \times [0,t]$.)

Actually, most of the linear equations have good stability properties(well-posedness).

Illustration on the well-posedness of linear equations

For the equation $L^*u = \delta(x)$, take Fourier transform and we derive $P(\xi)\hat{u}(\xi) = 1$, where P is a polynomial. Denote $\mathscr{F}^{-1}(\frac{1}{P})$ as G.

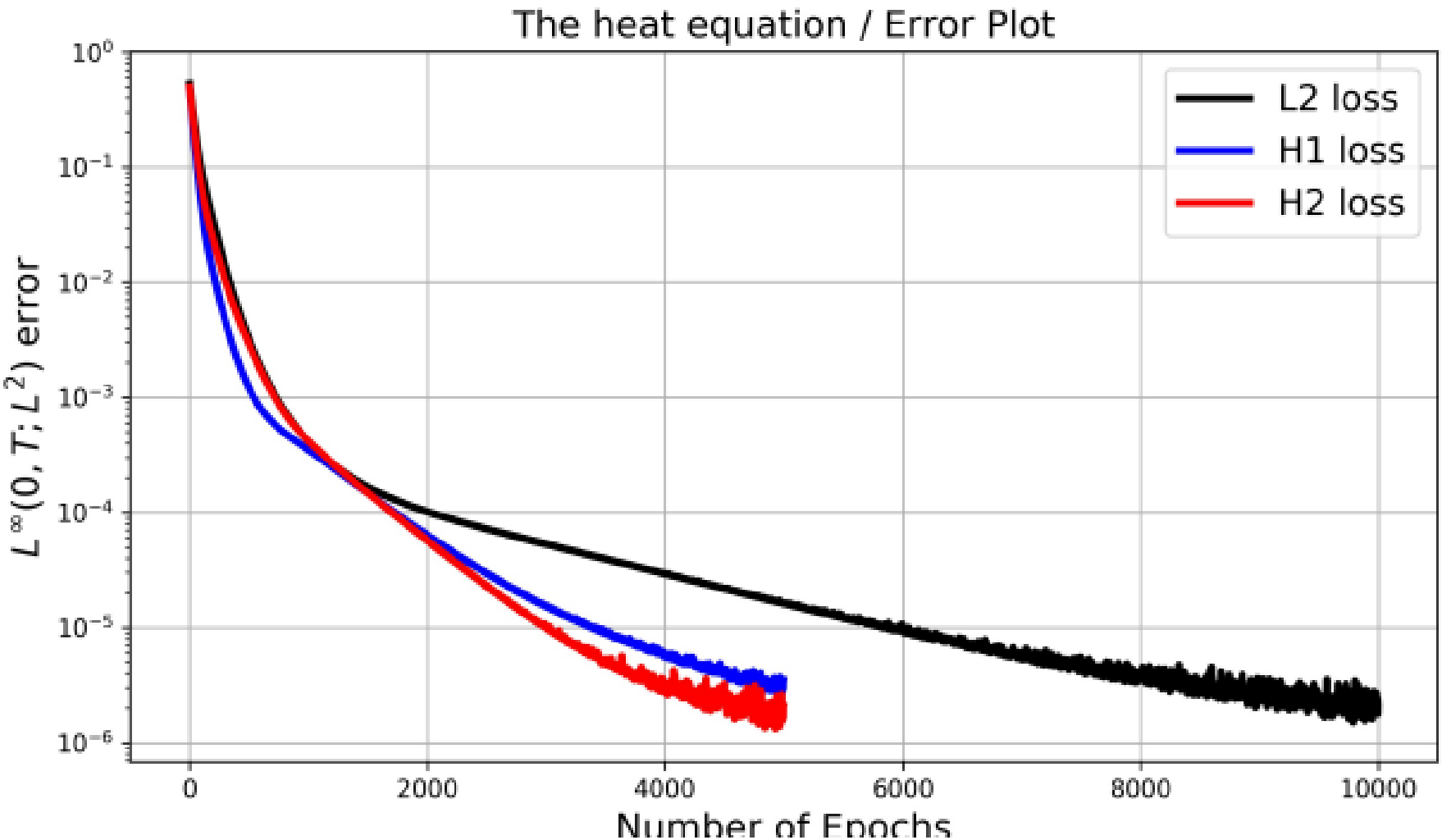
Formula

$$u(x') =< \delta(x-x'), u(x) >=< L^*G(x'-x), u >=< G, Lu >= G * f(x')$$

- Note that $D^k u = (D^k G) * f$.

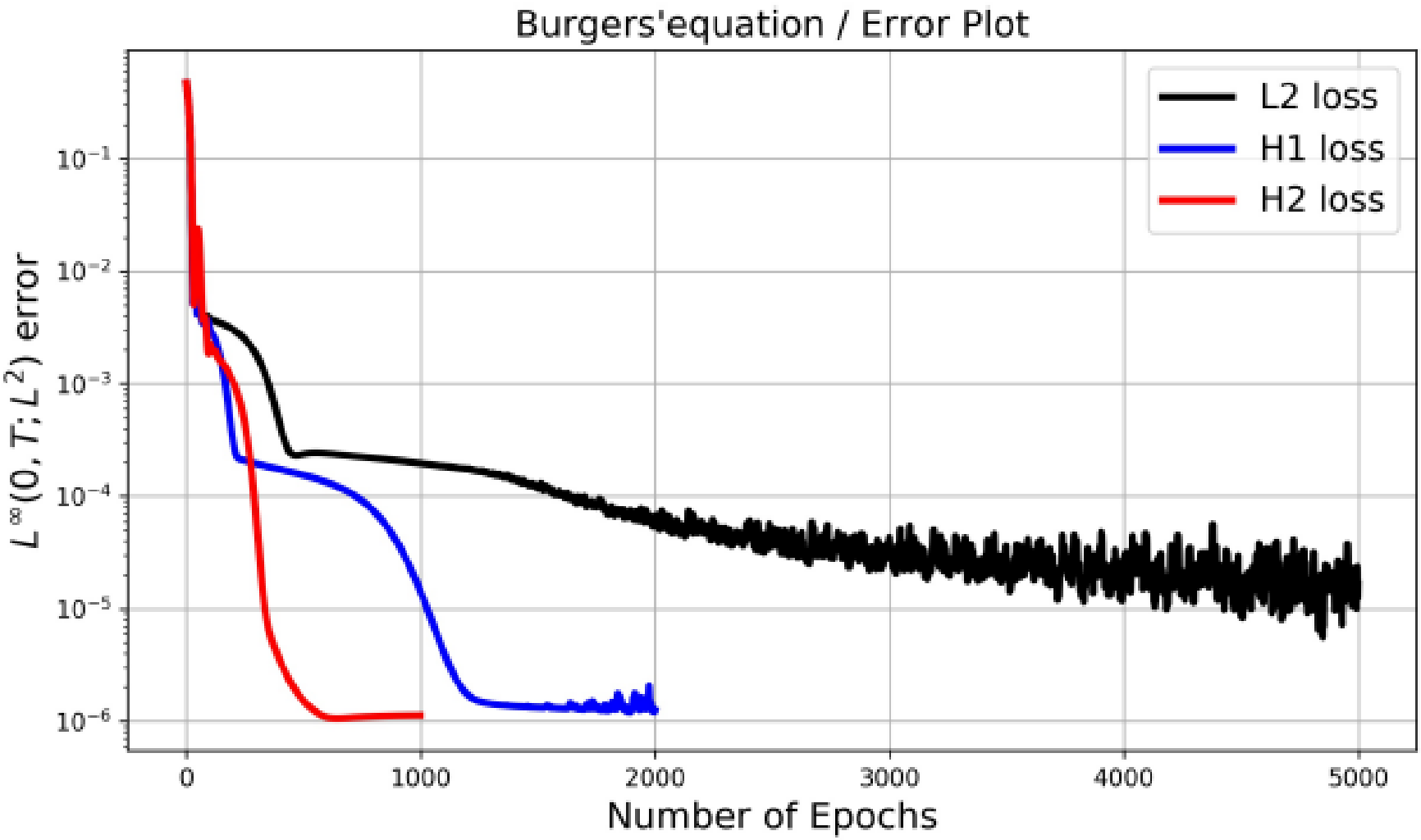
Less iteration for training

Numerical result in [3](ADAM).



Less iteration for training

Numerical result in [3](ADAM).



Necessity for high regularity loss term

For an equivalent form of a prototype of HJB Eqn.,

$$\begin{cases} \partial_t u - \Delta u + |Du|^2 = 0 \quad in\ \mathbb{R}^n \times [0,T] \\ u(x,0) = g(x). \end{cases}$$

Theorem

For any $\varepsilon > 0$, $A > 0$, $r \geq 1$, $m \in \mathbb{N}$ and $q \in [1, \frac{n}{4}]$, there exists a function u which has the same regularity as u_0(i.e. if u_0 belong to $W^{k,s}$ or C^k for some k, s, then so does u) such that $\|Lu\|_q < \varepsilon$, $Bu = Bu_0$ and $spt(u - u_0)$ is compact, while $\|u - u_0\|_{m,r} > A$.

u_0 : exact solution, L: inner operator, B: boundary operator.

Summary

(i) Could minimizing loss function with low regularity work?

In general, NO!

(ii) Would there be advantages to choose loss functions with high regularity?(e.g., higher accuracy, less time for training, etc)

Possibly, YES!

Summary

(i) Could minimizing loss function with low regularity work?

In general, NO!

(ii) Would there be advantages to choose loss functions with high regularity?(e.g., higher accuracy, less time for training, etc)

Possibly, YES!

How to minimize loss functions with high regularity (L^p norm with large p or $W^{m,p}$ norm with $m \geq 1$)?

Difficulties

(i) rounding error (the loss may be exact zero when p is large),

(ii) over-fitting,

(iii) much more sampling is necessary,

(iv) time-consuming to back-propagate,

(v) complicated landscape for optimization.

Method

Intuition

Note that the dual space of L^p is L^q for $p, q \in (1, \infty)$ and $\frac{1}{p} + \frac{1}{q} = 1$.

$$\forall u \in L^p,\ \|u\|_{L^p} = \|u\|_{L^{q^*}} = \sup_{v \neq 0} \frac{\int uv dx}{\|v\|_q} = \sup_{\|v\|_q \leq 1} \int uv dx = \sup_{\|v\|_2 \leq 1} \int |u|v|^{\frac{2}{q}}| dx$$

Thus, we can achieve small $L^p (p \to \infty)$ loss without facing L^p optimization.

Framework

Minimize *variational loss*$(:= \sup_{||v||_{X^*} \leq 1} < Lu - f, v >)$ w.r.t θ to solve the equation.

- X^*: the dual space of X.

Advantage 1

Framework

Minimize $\sup_{||v||_{X^*} \leq 1} < Lu - f, v >$ w.r.t θ to solve the equation.

- **Obtain better regularity by dealing with optimization problem consisting of lower regularity term.**

Advantage 2

Framework

Minimize $\sup_{||v||_{X^*} \leq 1} < Lu - f, v >$ w.r.t θ to solve the equation.

In fact, we are calculating the weak solution to the equation.

- **Achieve better robustness when the equation has no classical solution, or the solution has 'approximate singularity'.**

Remark

(i) Singularity is quite common in fluid equations.

Advantage 3

Framework

Minimize $\sup_{||v||_{X^*} \leq 1} < Lu - f, v >$ w.r.t θ to solve the equation.

Integral by parts ($< Lu - f, v > = < u, L^* v > - < f, v >$) so that the differential operator no longer operates on u.

- **This might partly overcome the difficulty in optimization.**

Remark

(i) Although there remains differential operator in the optimization of $\sup_{||v||_{X^*} \leq 1} < u, L^* v > - < f, v >$, we could apply some approximate methods to compute it (eg. Stein's Lemma).

(ii) This property holds only if L is of divergence form or L is linear.

Advantage 3

Remark

(i) Although there remains differential operator in the optimization of $\sup_{||v||_{X^*} \leq 1} < u, L^* v > - < f, v >$, we could apply some approximate methods to compute it (eg. Stein's Lemma).

(ii) This property holds only if L is of divergence form or L is linear.

Example

$\int_\Omega |Du|^2 v dx = \int_{\partial\Omega} uv\partial_n u dS - \int_\Omega uDu \cdot Dv dx - \int_\Omega u\Delta uv dx.$

VPINN(Variational PINN)[1]

Framework

(i) Choose a class of testing function $V = \{v_k\}_{k=1}^N$.

(ii) Inner loss$= \sum_{k=1}^N \langle Lu - f, v_k \rangle^2$.

(iii) Integral by parts to simplify the inner loss.

- In the paper, V is chosen as $\{sinkx\}_{k=1}^N$ or Legendre polinomial.
- Originally, VPINN stemmed from Galerkin method for solving linear PDEs in Hilbert space.

WAN(Weak Adversarial Network)[2]

Framework

(i) Train two NNs, u_θ for solution and ϕ_η for test function.

(ii) Inner loss $L_{int} = \log|\langle Lu_\theta - f, \phi_\eta\rangle|^2 - \log\|\phi_\eta\|_2^2$.

Total loss $\ell(\theta,\eta) = L_{int} + \alpha L_{bdry}(\theta)$

(iii) $\min_\theta \max_\eta \ell(\theta,\eta)$.

Algorithm 1 Weak Adversarial Network (WAN) for Solving High-dimensional static PDEs.

Input: N_r/N_b: number of region/boundary collocation points; K_u/K_φ: number of solution/adversarial network parameter updates per iteration.
Initialize: Network architectures $u_\theta, \varphi_\eta : \Omega \to \mathbb{R}$ and parameters θ, η.
while not converged **do**
  Sample collocation points $\{x_r^{(j)} \in \Omega : j \in [N_r]\}$ and $\{x_b^{(j)} \in \partial\Omega : j \in [N_b]\}$
  `# update weak solution network parameter`
  for $k = 1, \ldots, K_u$ **do**
    Update $\theta \leftarrow \theta - \tau_\theta \nabla_\theta L$ where $\nabla_\theta L$ is approximated using $\{x_r^{(j)}\}$ and $\{x_b^{(j)}\}$.
  end for
  `# update test function network parameter`
  for $k = 1, \ldots, K_\varphi$ **do**
    Update $\eta \leftarrow \eta + \tau_\eta \nabla_\eta L$ where $\nabla_\eta L$ is approximated using $\{x_r^{(j)}\}$.
  end for
end while
Output: Weak solution $u_\theta(\cdot)$ of (1).

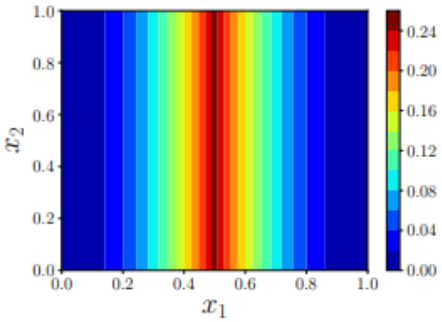
Experiments

Parameters:

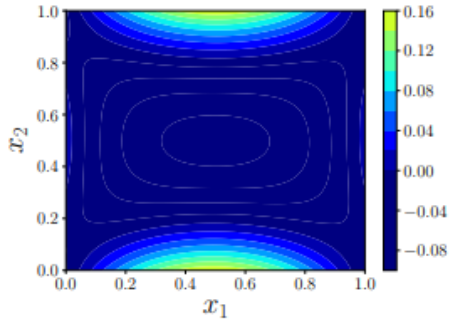
To quantitatively evaluate the accuracy of a solution u_θ, we use the L_2 relative error $\|u_\theta - u^*\|_2 / \|u^*\|_2$, where u^* is the exact solution of the problem and $\|u\|_2^2 = \int_\Omega |u|^2 \, dx$. To compute this error in high dimensional domain Ω, we use a regular mesh grid of size 100×100 for (x_1, x_2), and sampled one point x for each of these grid points (i.e., for each grid point (x_1, x_2), randomly draw values of $(x_3, \ldots, x_d)$ of x within the domain Ω). These points are sampled in advance and then used for all comparison algorithms to compute their errors. They are different from those sampled during training processes. In all experiments, we set both of the primal network (weak solution u_θ) and the adversarial network (test function φ_η) in the proposed algorithms as fully-connected feed-forward networks. Unless otherwise noted, u_θ network is set to have 6 hidden layers, with 40 neurons per hidden layer. The activation functions of u_θ are set to tanh for layers 1, 2, 4 and 6, and elu for the problem in Section 4.2.1 and softplus for all other problems for layers 3 and 5. We do not apply activation function in the last, output layer. For the network φ_η, it consists of 8 hidden layers with 50 neurons per hidden layer. The activation functions are set to tanh for layers 1 and 2, softplus for layers 3, 5, and 8, sinc for layers 2, 5, and 7, and again no activation in the last layer. The parameters θ and η of the

Use Adam optimizer for updating θ and AdaGrad optimizer for η.

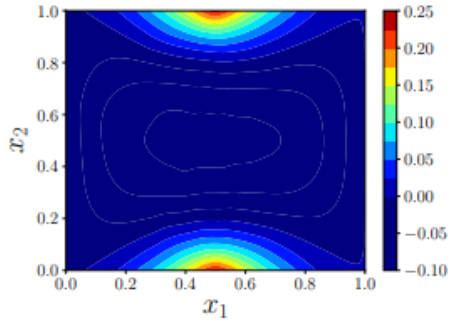
Results:Laplace Eqn in $(0,1)^2$



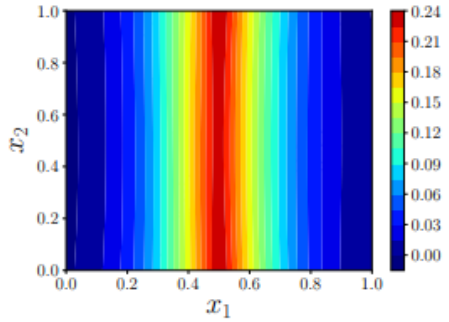
(a) True u^*



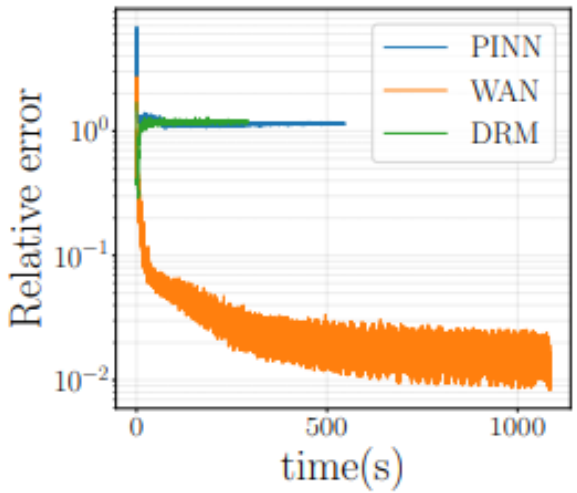
(b) u_{PINN}



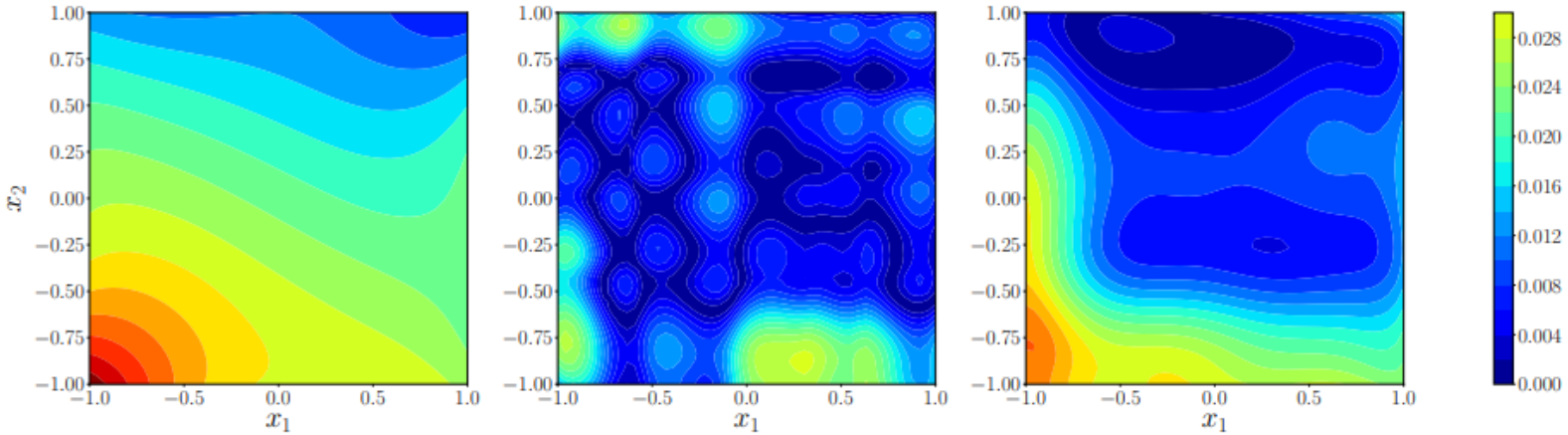
(c) u_{DRM}



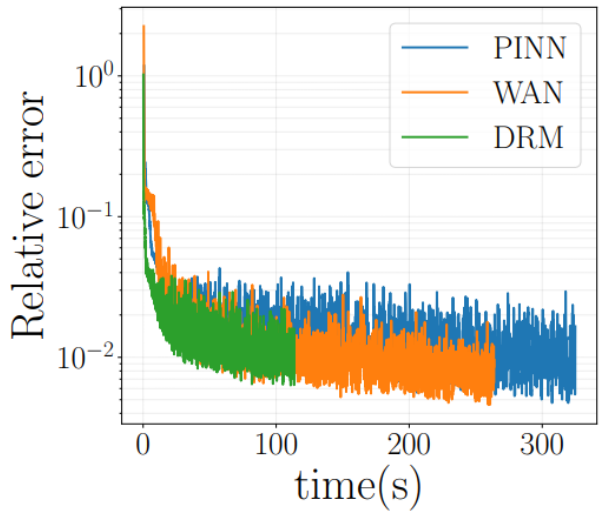
(d) u_{WAN}



Results:Laplace Eqn in $(0,1)^5$

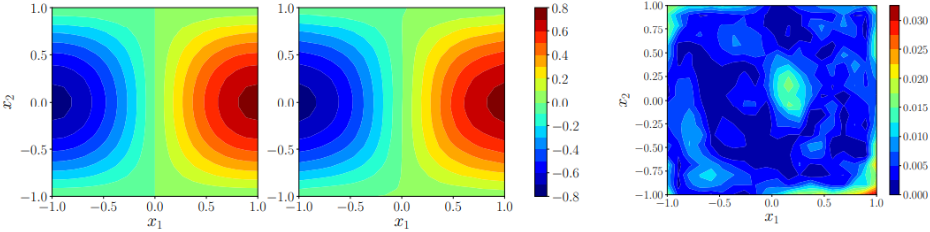


(a) $|u - u^*|$ with u obtained by PINN (left), DRM (middle), and WAN (right).



Results:parabolic equation involving time in $(0,1)^5, (0,1)^{10}$

$$
\begin{cases}
u_t - \Delta u - u^2 = f(x,t), & \text{in } \Omega \times [0,T] \\
u(x,t) = g(x,t), & \text{on } \partial\Omega \times [0,T] \\
u(x,0) = h(x), & \text{in } \Omega
\end{cases}
$$



(a) True $u^*(x,T)$ (left) vs estimated $u_\theta(x,T)$ (right)   (b) $|u_\theta(x,T) - u^*(x,T)|$

Recent work

- Domain decomposition (PFNN).
- Fitting boundary and interior respectively.

Future Direction

- Generalized version of WAN.
- Understand solutions as operators.
- Advance weak-form solver in $W^{m,p}$ or H^s.

1.Generalized version of WAN

Intuition

Note that the dual space of L^p is L^q for $p, q \in (1, \infty)$ and $\frac{1}{p} + \frac{1}{q} = 1$.

$$\forall u \in L^p,\ \|u\|_{L^p} = \|u\|_{L^{q^*}} = \sup_{v \neq 0} \frac{\int uv dx}{\|v\|_q} = \sup_{\|v\|_q \leq 1} \int uv dx = \sup_{\|v\|_2 \leq 1} \int |u| v|^{\frac{2}{q}}| dx$$

- q=2 corresponds to WAN.
- We could further try $\min_{\theta} \max_{\eta} \left\langle |Lu_\theta - f|, |v_\eta|^2 \right\rangle$.

2.Understanding solutions as operators

We could understand the solution u as functional: $v \to \int uvdx$.

framework

Simulate the functional $u(\cdot) : v \to \int uvdx$ with U_{NN}.

Inner residue $:= U_{NN}(L^* v_i) - \int f v_i dx$, L^* is the adjoint of L.

Possible advantages:

(i) In some cases, it is unnecessary to compute u pointwise(eg. u is a probability measure, and we are only interested in $u(X)$,i.e.$\mathbb{E}X$).

(ii) Achieve better robustness and faster convergence when the equation has no classical solution, or the solution has 'approximate singularity'.

(iii) Better generalization ability than existing operator methods (eg,Neural Operator).

Difficulty: How to handle nonlinear L ?

Example

Suppose there is a black box.

Input: a function v

Output: $\int uvdx$ for an unknown u.

Objective: Estimate $\int u^2 vdx$ for any input function v with the help of the black box.

Remark

Actually, we could query $v_i = \delta(x - x_i)$ and obtain $u(x_i)$ for a sufficiently dense set $\{x_i\}_{i=1}^N$. Obviously, this strategy is too expensive to be practical.

Could we try to learn the mapping: $u(\cdot) \to u^2(\cdot)$*?*

3.Advance weak-form solver in $W^{m,p}$ or H^s

Intuition

$\|Lu-f\|_X = \sup_{||v||_{X^*}\leq 1} \langle Lu-f, v\rangle$ and select $X = W^{m,p}$ or H^s.

$W^{m,p*} = W_0^{-m,q}$. $H^{s*} = H_0^{-s}$.

So far, loss term's high regularity on index m was beneficial yet unnecessary in applications. This might be the reason why $W^{m,p}$ weak-form solver has not come out yet.

However, we will show that loss term's high regularity in index m is necessary in eigenvalue problems.

Eigenvalue Problem

- Recall that in linear algebra, λ is an eigenvalue of a matrix $A \iff$ $\lambda I - A$ is invertible $\iff$ there exist $x \neq 0$ such that $(\lambda I - A)x = 0$
- We could generalize the definition of eigenvalue as follow:

Definition

For an operator A, suppose there is a class of operators $\{A(\lambda)\}$ constructed with A, where λ is parameter in the operator. We refer to λ_0 as eigenvalue if $A(\lambda_0)$ has distinctive property.

Eigenvalue Problem

Difficulty

The existence of trivial solution brings about an inevitable difficulty for PINN to solve eigenvalue problem.

Example

Suppose we know λ_0 is an eigenvalue of $-D^2$ in $H_0^2([-1,1])$ and want to give corresponding eigenfunction.

PINN: minimize $\|u''+\lambda_0 u\|^2_{L^2(\Omega)}+\mu||u||^2_{L^2(\partial\Omega)}$

Unfortunately, $u=0$ is also a minimizer.

Further constraining $u(0)=1$ and adding an loss term $|u(0)-1|^2$ might not help, because $\mathbf{1}_{x=0}$ is a minimizer.

Eigenvalue Problem

Self-similar blow-up profile for the Boussinesq equations via a physics-informed neural network

Yongji Wang[*], Ching-Yao Lai[†], Javier Gómez-Serrano[‡], Tristan Buckmaster[§]

eigenvalue problem

Self-similar Burgers' Equation $-\lambda u + ((1+\lambda)x + u)\frac{du}{dx} = 0$. (Denote as $L(\lambda)u = 0$).

Constrain $u(-2) = 1$.

(Ground Truth): $x = -u - |u|^{1+\frac{1}{\lambda}}$

$u(x)$ is smooth $\iff$ $\lambda = \frac{1}{2i+2}, i = 0, 1, 2, ..$

Experiment

- Initialize $\lambda \in [\frac{1}{2i+3}, \frac{1}{2i+1}]$.
- Optimize $\min_{\lambda,\theta} ||L(\lambda)u_\theta||^2_{H^1} + \mu(u_\theta(-2)-1)^2$.
- Surprisingly, λ converges to $\frac{1}{2i+2}$.
- It is remarkable since traditional numerical methods could only get stable eigenvalue ($\frac{1}{2}, i = 0$ *here*).

References

1. Ehsan Kharazmi, Zhongqiang Zhang, George Em Karniadakis
   Variational physics-informed neural networks for solving partial differential equations
   arXiv preprint arXiv:1912.00873
2. Yaohua Zanga, Gang Bao, Xiaojing Ye, Haomin Zhou
   Weak adversarial networks for high-dimensional partial differential equations
   Journal of Computational Physics, 2020
3. Hwijae Son, Jin Woo Jang, Woo Jin Han, Hyung Ju Hwang
   Sobolev training for the neural network solutions of pdes
   arXiv preprint arXiv:2101.08932

References

1. Yongji Wang, Ching-Yao Lai, Javier Gomez-Serrano, Tristan Buckmaster

   Self-similar blow-up profile for the Boussinesq equations via a physics-informed neural network

   arXiv preprint arXiv:2201.06780

Thanks!