L^2 is not the right loss for PINN when solving high dimensional nonlinear PDE

Chuwei Wang, Shanda Li, Di He, Liwei Wang
Peking University

Outline

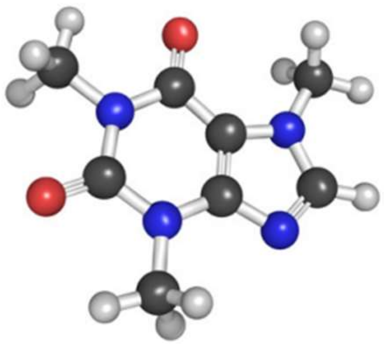
1. Introduction
2. Definition of Stability
3. Bounds on the Stability of PINN for HJB Equation
4. New Algorithms
5. Conclusion & Future Direction

Outline

1. Introduction
2. Theoretical Analysis for the Validity of PINN
3. Failure of PINN for High Dimensional HJB Equation
4. New Algorithm for High Dimensional HJB Equation
5. Conclusion & Future Direction

Preliminary: Partial Differential Equation

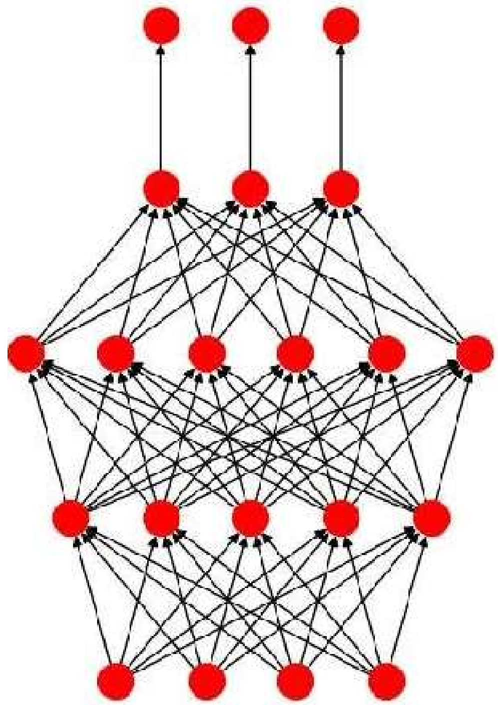
Partial Differential Equation (PDE) is a ubiquitous tool in mathematical modeling of physics, control, and finance.



- Solving PDE is important for understanding these systems.
- Designing an accurate and efficient PDE solver is very challenging.

Preliminary: Deep Learning

Deep Learning has achieved huge successes in computer vision, natural language processing, and graph-based learning.



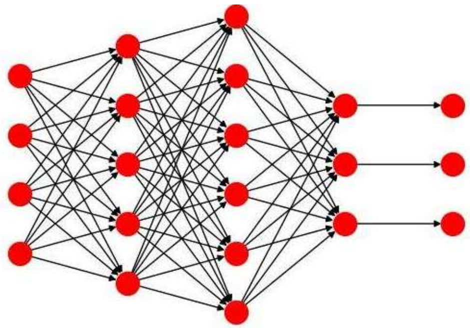
Can we use deep learning to solve PDE?

+

=

$$\begin{cases} \mathcal{L}u(x) = \varphi(x) \\ \mathcal{B}u(x) = \psi(x) \end{cases}$$

+



=

?

Formulation of Partial Differential Equation

Partial Differential Equation involves an unknown multi-variable function $u(x)$ and partial derivatives of the unknown function.

$$\begin{cases} \mathcal{L}u(x) = \varphi(x) & x \in \Omega \subset \mathbb{R}^n \\ \mathcal{B}u(x) = \psi(x) & x \in \partial\Omega, \end{cases}$$

$\mathcal{L}$: partial differential operator.
B: boundary condition.

PINN: solving PDE with deep learning

Physics-informed Neural Networks (PINN):

- Solving PDE as a function approximate problem.
- Training an NN to express the PDE solution with L^2 **Physics-Informed Loss**.

$$\begin{cases} \mathcal{L}u(x) = \varphi(x) \\ \mathcal{B}u(x) = \psi(x) \end{cases} \Rightarrow \quad \begin{aligned} \ell_{\Omega}(u) &= \|\mathcal{L}u(x) - \varphi(x)\|_{L^2(\Omega)}^2, \\ \ell_{\partial\Omega}(u) &= \|\mathcal{B}u(x) - \psi(x)\|_{L^2(\partial\Omega)}^2. \end{aligned}$$

Neural Network: $u_\theta(x)$ with x as the input and θ as the parameters.

PINN is straightforward and successful.

Can we use it to solve **high-dimensional PDEs**?

- Conventional methods fail due to the **curse of dimensionality**.
- Neural networks do well in representing **high-dimensional mappings**.

PINN is straightforward and successful.

Can we use it to solve **high-dimensional PDEs**?

- PINN's **accuracy** is not satisfactory on high-dimensional non-linear PDEs.

Outline

1.Introduction

2. Definition of Stability

3. Failure of PINN for High Dimensional HJB Equation

4. New Algorithm for High Dimensional HJB Equation

5. Conclusion & Future Direction

Theoretical Analysis for the Validity of PINN

$$\ell_{\Omega}(u) = \|\mathcal{L}u(x) - \varphi(x)\|^2_{L^2(\Omega)},$$

$$\ell_{\partial\Omega}(u) = \|\mathcal{B}u(x) - \psi(x)\|^2_{L^2(\partial\Omega)}.$$

- PINN uses L^2 Physics-Informed Loss by default.

  Zero Training Loss $\Leftrightarrow$ Learned solution is exactly accurate

- But practically we only obtain **small** but **non-zero** losses.

*Does **a learned solution with a small loss** always corresponds to **a good approximator of the exact solution**?*

A closer look at the learned solution

A learned solution $u_\theta(x)$ is the solution to **a *perturbed* PDE**:

$$\begin{cases} \mathcal{L}u(x) = \varphi(x) + (\mathcal{L}u_\theta(x) - \varphi(x)) & x \in \Omega \subset \mathbb{R}^n \\ \mathcal{B}u(x) = \psi(x) + (\mathcal{B}u_\theta(x) - \psi(x)) & x \in \partial\Omega \end{cases}$$

The scale of the perturbation can be characterized by the **Physics-Informed Loss**:

$$\ell_\Omega(u) = \|\mathcal{L}u(x) - \varphi(x)\|^2_{L^2(\Omega)},$$

$$\ell_{\partial\Omega}(u) = \|\mathcal{B}u(x) - \psi(x)\|^2_{L^2(\partial\Omega)}.$$

Stability of PDEs

The accuracy of PINN is closely related to the ***stability*** of PDE.

In PDE literature, we say an equation is ***stable*** if the solution of the perturbed PDE converges to the exact solution as the perturbations approach zero (measured by certain norm).

$$\begin{cases} \mathcal{L}u(x) = \varphi(x) + (\mathcal{L}u_\theta(x) - \varphi(x)) \\ \mathcal{B}u(x) = \psi(x) + (\mathcal{B}u_\theta(x) - \psi(x)) \end{cases} \Rightarrow \begin{cases} \mathcal{L}u(x) = \varphi(x) \\ \mathcal{B}u(x) = \psi(x) \end{cases}$$

Approximation　　　　　　　　　　Ground truth

Stability of PDEs

$$\begin{cases} \mathcal{L}u(x) = \varphi(x) & x \in \Omega \subset \mathbb{R}^n \\ \mathcal{B}u(x) = \psi(x) & x \in \partial\Omega, \end{cases}$$

[Definition] We say a PDE is (Z_1, Z_2, Z_3)-**stable**, if

$$\|u^*(x) - u(x)\|_{Z_3} = O(\|\mathcal{L}u(x) - \varphi(x)\|_{Z_1} + \|\mathcal{B}u(x) - \psi(x)\|_{Z_2})$$

as $\|\mathcal{L}u(x) - \varphi(x)\|_{Z_1}, \|\mathcal{B}u(x) - \psi(x)\|_{Z_2} \to 0$, where Z_1, Z_2, Z_3 are three Banach spaces and u^* is the exact solution.

- Loss functions corresponding to $||\cdot||_{Z_1}$ and $||\cdot||_{Z_2}$ help to obtain u_θ that is ***provably*** close to the exact solution.

Stability of PDEs

PEKING UNIVERSITY

[Definition] We say a PDE is (Z_1, Z_2, Z_3)-**stable**, if

$$\|u^*(x) - u(x)\|_{Z_3} = O(\|\mathcal{L}u(x) - \varphi(x)\|_{Z_1} + \|\mathcal{B}u(x) - \psi(x)\|_{Z_2})$$

as $\|\mathcal{L}u(x) - \varphi(x)\|_{Z_1}, \|\mathcal{B}u(x) - \psi(x)\|_{Z_2} \to 0$, where Z_1, Z_2, Z_3 are three Banach spaces and u^* is the exact solution.

- PINN training with ***L^2* Physics-Informed Loss** is suitable only when a PDE is (L^2, L^2, Z)-stable for some Banach space Z.

$$\ell_\Omega(u) = \|\mathcal{L}u(x) - \varphi(x)\|^2_{L^2(\Omega)},$$

$$\ell_{\partial\Omega}(u) = \|\mathcal{B}u(x) - \psi(x)\|^2_{L^2(\partial\Omega)}.$$

Outline

[Theorem1] (informal) A large class of n-dimensional HJB Equation is $(L^p, L^q, W^{1,1})$-stable if $p>n$, $q>kn$ (k depends on the equation).

[Theorem2] (informal) A large class of n-dimensional HJB Equation is *not* $(L^p, L^q, W^{1,1})$-stable if $p<n/4$.

HJB Equation: **H**amilton-**J**acobi-**B**ellman Equation

- The class of PDE we study is representative in high-dimensional non-linear PDEs. Power-law trading cost in optimal execution problem, Linear-Quadratic-Gaussian control and Merton's portfolio model are all special cases of this form.

$$\begin{cases} \mathcal{L}_{\mathrm{HJB}} u := \partial_t u(x,t) + \frac{1}{2}\sigma^2 \Delta u(x,t) - \sum_{i=1}^{n} A_i |\partial_{x_i} u|^{c_i} = \varphi(x,t) & (x,t) \in \mathbb{R}^n \times [0,T] \\ \mathcal{B}_{\mathrm{HJB}} u := u(x,T) = g(x) & x \in \mathbb{R}^n \end{cases}$$

- We consider $W^{1,1}$-stability here because both u and ∇u is important in application.

Theoretical analysis

[Theorem1] (informal) A large class of n-dimensional HJB Equation is $(L^p, L^q, W^{1,1})$-stable if $p>n$, $q>kn$ (k depends on the equation).

Theorem 4.3. *For $p, q \geq 1$, let $r_0 = \frac{(n+2)q}{n+q}$. Assume the following inequalities hold for p, q and r_0:*

$$p \geq \max\left\{2, \left(1 - \frac{1}{\bar{c}}\right) n\right\};\ q > \frac{(\bar{c}-1)n^2}{(2-\bar{c})n+2};\ \frac{1}{r_0} \geq \frac{1}{p} - \frac{1}{n}, \tag{7}$$

where $\bar{c} = \max_{1 \leq i \leq n} c_i$ in Eq. (6). Then for any $r \in [1, r_0)$ and any bounded open set $Q \subset \mathbb{R}^n \times [0, T]$, Eq. (6) is $(L^p(\mathbb{R}^n \times [0,T]), L^q(\mathbb{R}^n), W^{1,r}(Q))$-stable for $\bar{c} \leq 2$.

Theoretical analysis

[Theorem1] (informal) A large class of n-dimensional HJB Equation is (L^p, L^q, $W^{1,1}$)-stable if $p>n$, $q>kn$ (k depends on the equation).

Theorem 4.3. *For $p, q \geq 1$, let $r_0 = \frac{(n+2)q}{n+q}$. Assume the following inequalities hold for p, q and r_0:*

$$p \geq \max\left\{2, \left(1 - \frac{1}{\bar{c}}\right) n\right\};\ q > \frac{(\bar{c}-1)n^2}{(2-\bar{c})n+2};\ \frac{1}{r_0} \geq \frac{1}{p} - \frac{1}{n}, \tag{7}$$

where $\bar{c} = \max_{1 \leq i \leq n} c_i$ in Eq. (6). Then r any $r \in [1, r_0)$ and ny bounded open set $Q \subset \mathbb{R}^n \times [0, T]$, Eq. (6) is $(L^p(\mathbb{R}^n \times [0,T]), L^q(\mathbb{R}^n)$, $V^{1,r}(Q))$-stable for ≤ 2.

$\mathrm{O}(n)$ $\mathrm{O}(n)$

Theoretical analysis

[Theorem2] (informal) A large class of n-dimensional HJB Equation is *not* $(L^p, L^q, W^{1,1})$-stable if $p<n/4$.

Theorem 4.4. *There exists an instance of Eq. (6), whose exact solution is u^*, such that for any $\varepsilon > 0, A > 0, r \geq 1, m \in \mathbb{N}$ and $q \in \left[1, \frac{n}{4}\right]$, there exists a function $u \in C^\infty(\mathbb{R}^n \times (0,T])$ which satisfies the following conditions:*

- $\|\mathcal{L}_{\mathrm{HJB}}u - \varphi\|_{L^q(\mathbb{R}^n\times[0,T])} < \varepsilon$, $\mathcal{B}_{\mathrm{HJB}}u = \mathcal{B}_{\mathrm{HJB}}u^*$, *and* $\mathrm{supp}(u-u^*)$ *is compact, where* $\mathcal{L}_{\mathrm{HJB}}$ *and* $\mathcal{B}_{\mathrm{HJB}}$ *are defined in Eq. (6).*
- $\|u - u^*\|_{W^{m,r}(\mathbb{R}^n\times[0,T])} > A$.

Theoretical analysis

[Theorem2] (informal) A large class of n-dimensional HJB Equation is *not* (L^p, L^q, $W^{1,1}$)-stable if $p<n/4$.

Theorem 4.4. *There exists an instance of Eq. (6), whose exact solution is u^*, such that* ***for any*** $\varepsilon > 0, A > 0, r \geq 1, m \in \mathbb{N}$ *and* $p \in \left[1, \frac{n}{4}\right]$, *there* ***exists a function*** $u \in C^\infty(\mathbb{R}^n \times (0, T])$ *which satisfies the following conditions:*

- $\|\mathcal{L}_{\text{HJB}} u - \varphi\|_{L^p(\mathbb{R}^n \times [0,T])} < \varepsilon$, $\mathcal{B}_{\text{HJB}} u = \mathcal{B}_{\text{HJB}} u^*$, *and* $\text{supp}(u - u^*)$ *is compact, where* $\mathcal{L}_{\text{HJB}}$ *and* $\mathcal{B}_{\text{HJB}}$ *are defined in Eq. (6).*
- $\|u - u^*\|_{W^{m,r}(\mathbb{R}^n \times [0,T])} > A$.

- Set $m = 0$, then Sobolev norm becomes L^r-norm.
- The distance between u_θ and u^*, ∇u_θ and ∇u^* could be **arbitrarily large** even though the ***L^2 loss* is small**!

Empirical results (100-dimensional HJB)

Table 6: **Error/loss-vs-time result of original PINN for Eq. (12).**

Iteration	**1000**	**2000**	**3000**	**4000**	**5000**
L^2 Loss	0.098	0.088	0.070	0.584	0.041
L^1 Relative Error	6.18%	5.36%	3.86%	3.94%	3.47%
$W^{1,1}$ Relative Error	17.53%	17.67%	14.83%	14.40%	11.31%

- L^2 loss drops very quickly, while relative error remains high.

Outline

Recap: theoretical analysis

[Theorem1] (informal) A large class of n-dimensional HJB Equation is $(L^p, L^q, W^{1,1})$-stable if $p>n,\ q>kn$ (k depends on the equation).

[Theorem2] (informal) A large class of n-dimensional HJB Equation is *not* $(L^p, L^q, W^{1,1})$-stable if $p<n/4$.

L^2 Loss is not suitable for high-dimensional HJB Equation.

***L^p Loss (p>>1 or $p=\infty$)** can be a better choice!*

Experiments: Naïvely minimizing L^p loss

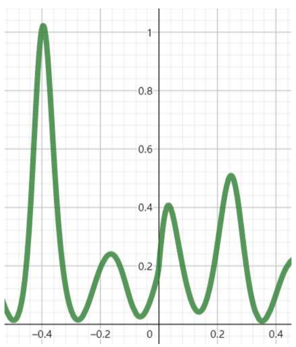
- Naïvely minimizing L^p loss with large but finite p does not lead to satisfactory results.

Method	Error	
	Domain	Boundary
L^4 Loss	2.42%	13.64%
L^8 Loss	53.55%	23.78%
L^{16} Loss	113.24%	80.68%

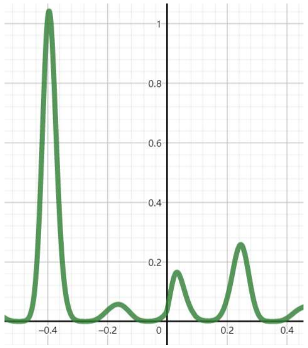
Experiments: Naïvely minimizing L^p loss

- Naïvely minimizing L^p loss with large but finite p does not lead to satisfactory results.
- Possible reasons:

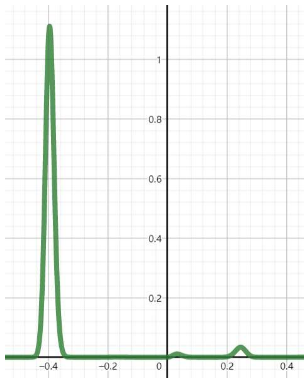
Large p ➡ { Large MC variance ➡ Sampling inefficiency; Stiff landscape ➡ Optimization difficulty }



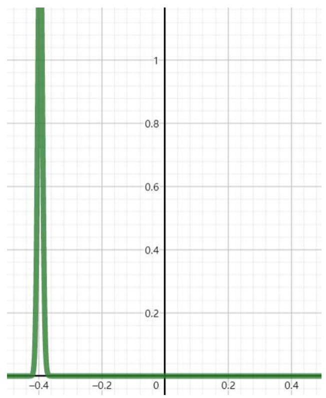
p =1



p =2



p =5



p =20

Minimizing L^∞ Physics-Informed Loss

New training objective: L^∞ Physics-Informed Loss

$$\ell_\infty(u) = \sup_{x\in\Omega} |\mathcal{L}u(x) - \varphi(x)| + \lambda \sup_{x\in\partial\Omega} |\mathcal{B}u(x) - \psi(x)|$$

Algorithm: adversarial-training-like min-max optimization.

- Inner loop: gradient-based methods to obtain data points with large point-wise loss to approximate supremum.
- Outer loop: fix the generated data points and calculate the gradient g to learn the network parameters.

L^∞ training for Physics-Informed Neural Networks

Algorithm 1 L^∞ Training for Physics-Informed Neural Networks

Input: Target PDE (Eq. (1)); neural network u_θ; initial model parameters θ
Output: Learned PDE solution u_θ
Hyper-parameters: Number of total training iterations M; number of iterations and step size of inner loop K, η; weight for combining the two loss term λ

1: **for** $i = 1, \cdots, M$ **do**
2: Sample $x^{(1)}, \cdots, x^{(N_1)} \in \Omega$ and $\tilde{x}^{(1)}, \cdots, \tilde{x}^{(N_2)} \in \partial\Omega$
3: **for** $j = 1, \cdots, K$ **do**
4: **for** $k = 1, \cdots, N_1$ **do**
5: $x^{(k)} \leftarrow \text{Project}_\Omega\left(x^{(k)} + \eta \,\text{sign}\nabla_x \left(\mathcal{L}u_\theta(x^{(k)}) - \varphi(x^{(k)})\right)^2\right)$
6: **for** $k = 1, \cdots, N_2$ **do**
7: $\tilde{x}^{(k)} \leftarrow \text{Project}_{\partial\Omega}\left(\tilde{x}^{(k)} + \eta \,\text{sign}\nabla_x \left(\mathcal{B}u_\theta(\tilde{x}^{(k)}) - \psi(\tilde{x}^{(k)})\right)^2\right)$
8: $g \leftarrow \nabla_\theta\left(\frac{1}{N_1}\sum_{i=1}^{N_1}\left(\mathcal{L}u_\theta(x^{(i)}) - \varphi(x^{(i)})\right)^2 + \lambda \cdot \frac{1}{N_2}\sum_{i=1}^{N_2}\left(\mathcal{B}u_\theta(\tilde{x}^{(i)}) - \psi(\tilde{x}^{(i)})\right)^2\right)$
9: $\theta \leftarrow \text{Optimizer}(\theta, g)$
10: **return** u_θ

3-7:computing supremum (gradient ascend for data points)

8-9: optimization (gradient descent for NN parameters)

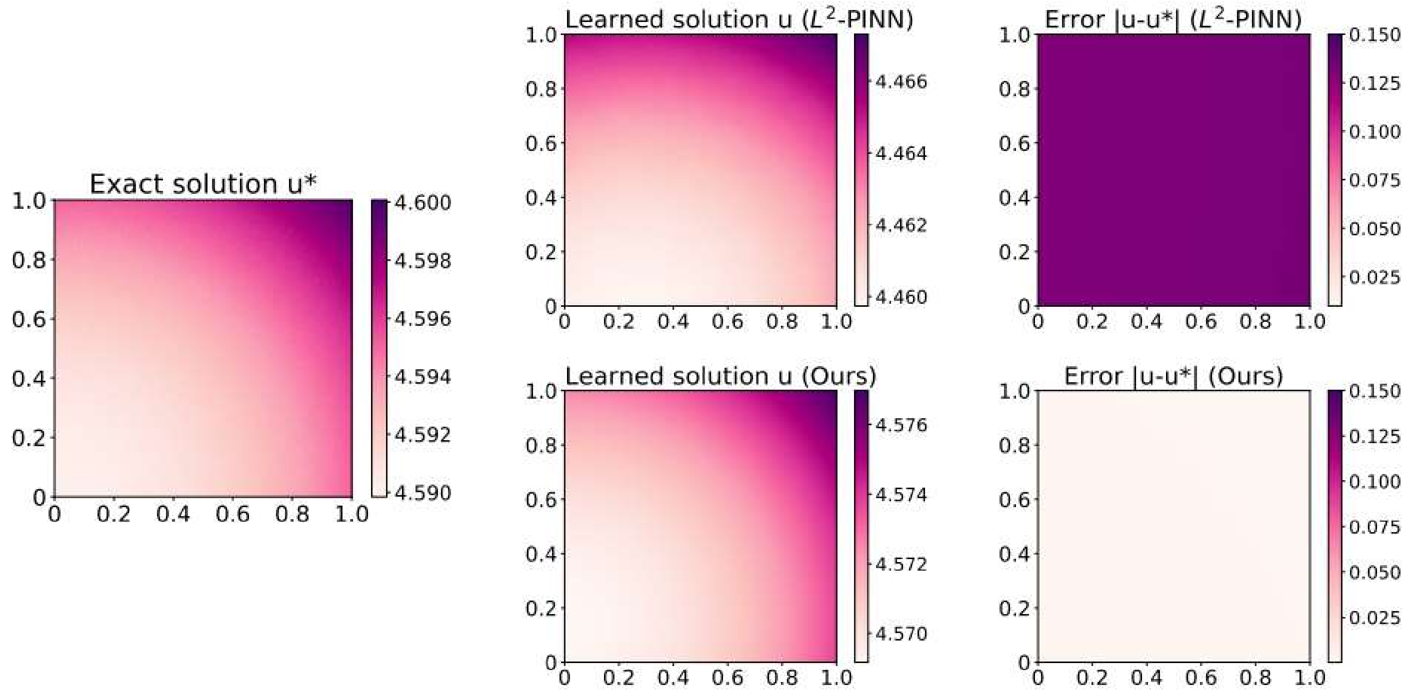
Experiments: High-dimensional HJB Equation

Method	$n = 100$		$n = 250$	
	Domain	Boundary	Domain	Boundary
Original PINN [23]	3.47%	19.59%	6.74%	23.25%
Adaptive time sampling [30]	3.05%	15.37%	7.18%	23.66%
Learning rate annealing [29]	11.09%	17.73%	6.94%	25.10%
Curriculum regularization [15]	3.40%	16.41%	6.72%	22.67%
Adversarial training (ours)	**0.27%**	**0.63%**	**0.95%**	**0.48%**

10x more accurate compared with baseline methods!

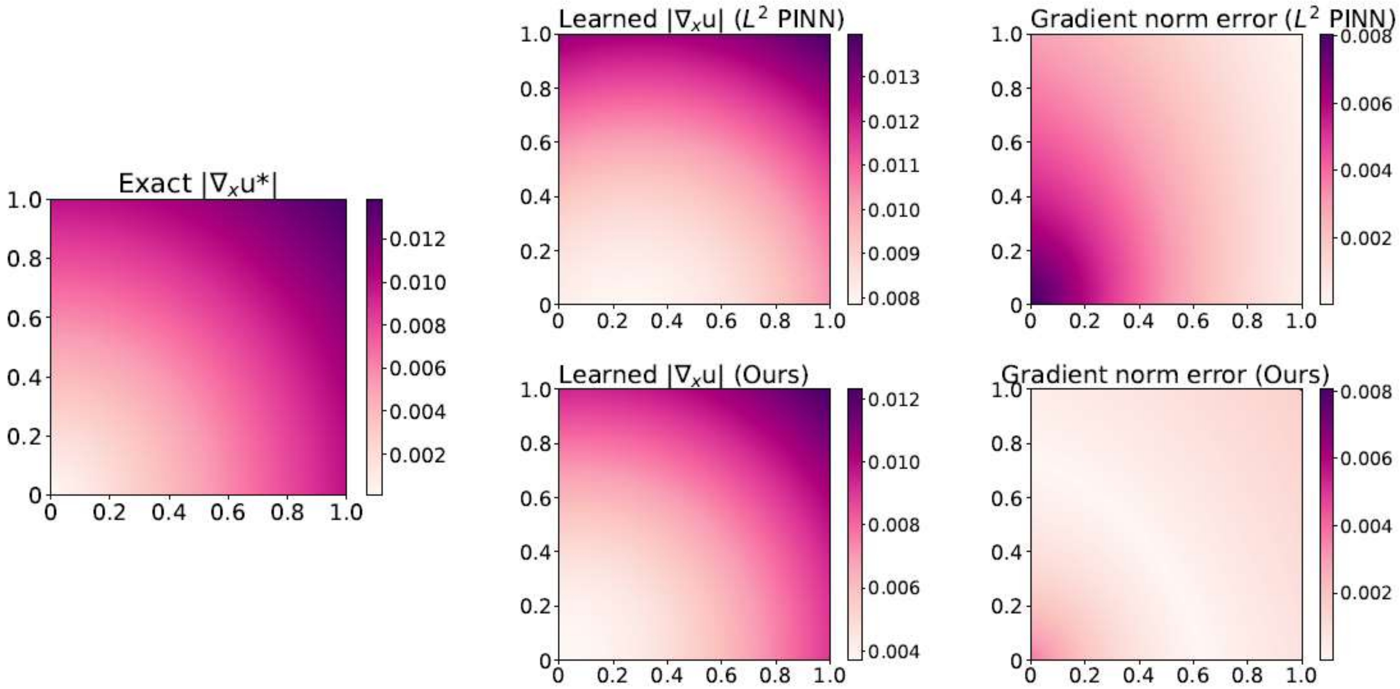
Experiments: High-dimensional HJB Equation

- Visualization of the learned solution of PINN and our method.



Experiments: High-dimensional HJB Equation

- Visualization of the ***gradient*** norm of the learned solution of PINN and our method.



Outline

1. Introduction

2. Theoretical Analysis for the Validity of PINN

3. Failure of PINN for High Dimensional HJB Equation

4. New Algorithm for High Dimensional HJB Equation

5. Conclusion & Future Direction

Conclusion

- In our work, we prove that for general L^p loss function, it is suitable for high dimensional HJB equation only if p is sufficiently large.
- Based on the theoretical results, we propose a novel PINN training algorithm to minimize the L^∞ loss for HJB equation in a similar spirit to adversarial training.

Future Direction

- Analyzing the stability properties of other important PDEs.
- Designing more efficient algorithms of L^∞ training for Physics-Informed Neural Networks.


Thanks!
北京大學
PEKING UNIVERSITY
paper can be found at https://arxiv.org/abs/2206.02016