Is L^2 Physics-Informed Loss Always Suitable for Training Physics-Informed Neural Network?

Chuwei Wang*
Shanda Li*
Di He
Liwei Wang

PEKING UNIVERSITY
Carnegie Mellon University
MACHINE LEARNING DEPARTMENT
NEURAL INFORMATION PROCESSING SYSTEMS

Background

Partial Differential Equation (PDE)

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

$\mathcal{L}$: PDE operator
$\mathcal{B}$: boundary/initial conditions
x: spatiotemporal-dependent variable

Physics-Informed Neural Network (PINN)

Train an NN, u_θ, to represent the PDE solution by minimizing the *Physics-Informed Loss*:

$$\ell_{\Omega,p}(u) = \|\mathcal{L}u(x) - \varphi(x)\|^p_{L^p(\Omega)},$$
$$\ell_{\partial\Omega,p}(u) = \|\mathcal{B}u(x) - \psi(x)\|^p_{L^p(\partial\Omega)}.$$

- $p = 2$ for vanilla PINN.

Probing PINN training

- Zero training loss ⇔ Learned solution is exactly accurate
- Small but non-zero loss ⇒ **?** (This work)

Does a learned solution with a small Physics-Informed Loss always corresponds to a good approximator of the exact solution?

No!

Hamilton-Bellman-Jacobi (HJB) Equation

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

- Most important PDE in stochastic control.
- Representative of high-dimensional nonlinear PDE.

Empirical finding: Relative error remains large in spite of the small training loss.

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%

PINN through the lens of stability

Stability of PDE

[Definition] 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.

- Physics-Informed loss functions corresponding to $\|\cdot\|_{Z_1}$ and $\|\cdot\|_{Z_2}$ help to obtain u_θ which is ***provably*** close to 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*.

Stability property of HJB Equation

[Theorem 1] (Stability of HJB equations)
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 2] (Instability of HJB equations)
There exists an instance of HJB Equation, whose exact solution is u^*, such that **for any** $\varepsilon > 0, A > 0, r \geq 1, m \in \mathbf{N}$ and $p \in [1, n/4]$, there exists a smooth function u which satisfies

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

- We characterize the error using Sobolev norm as the gradient of the solution is also important for HJB Equation.

The distance between u_θ and u^*, ∇u_θ and ∇u^* can be arbitrarily large even though the L^2 loss is small!

Minimizing L^∞ loss using adversarial training

New training objective for PINN

Theoretical results inspires us to minimize 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 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

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

Inner loop: ***Gradient ascend for data points***

Outer loop: ***Gradient descend for NN parameters***

Experiments

High-dimensional LQG Problem

$$\begin{cases} \partial_t u(x,t) + \Delta u(x,t) - \mu\|\nabla_x u(x,t)\|^2 = 0 \\ u(x,T) = g(x) \end{cases}$$

Method	Relative error for $n = 100$ L^1	L^2	$W^{1,1}$
Original PINN [28]	3.47%	4.25%	11.31%
Adaptive time sampling [35]	3.05%	3.67%	13.63%
Learning rate annealing [34]	11.09%	11.82%	33.61%
Curriculum regularization [17]	3.40%	3.91%	9.53%
Adversarial training (ours)	**0.27%**	**0.33%**	**2.22%**

More HJB Equation variants

Method	$c = 1.25$	$c = 1.5$	$c = 1.75$
Original PINN [28]	1.11%	3.82%	2.73%
Adaptive time sampling [35]	1.18%	2.34%	7.94%
Learning rate annealing [34]	0.98%	1.13%	1.06%
Curriculum regularization [17]	6.27%	0.37%	3.51%
Adversarial training (ours)	**0.61%**	**0.15%**	**0.29%**

Solution visualization