Machine Learning with Physics

Scaling Law and Minimax Optimality

Yiping Lu

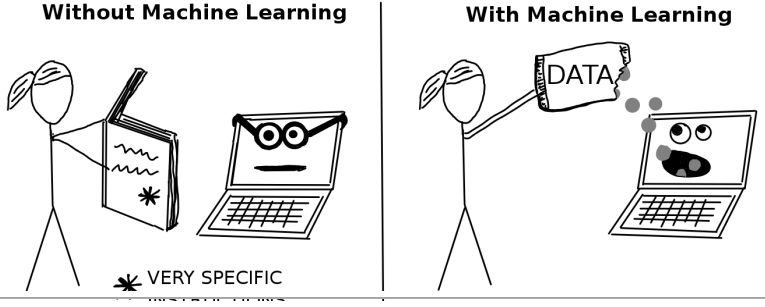
Institute for Computational and Mathematical Engineering School Of Engineering
Stanford University

Joint work with Haoxuan Chen, Jianfeng Lu, Lexing Ying and Jose Blanchet.

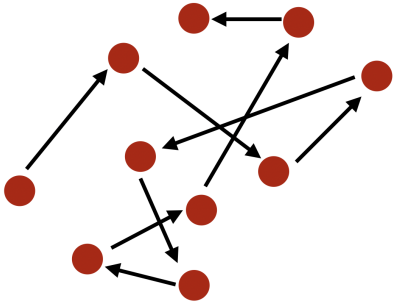
Motivation

We can make Predictions from

- physics using PDEs/Structure Form
- data using Machine Learning



Examples



$$dX_t = f(X_t)dt + dW_t$$

We can estimate the drift from a single path of diffusion via solving the PDE

$$\frac{1}{2}\Delta\mu - f.\nabla\mu - div(f)\mu = 0,$$

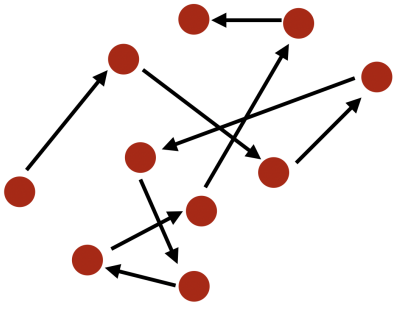
from random sample of μ.

[1] Maximum-likelihood estimation for diffusion processes via closed-form density expansions. AOS, 2013.

[2] Nonparametric statistical inference for drift vector fields of multi-dimensional diffusions. AOS, 2020.

[3] Semiparametric estimation of McKean-Vlasov SDEs, arXiv:2107.00539.

Examples



$$dX_t = f(X_t)dt + dW_t$$

We can estimate the drift from a single path of diffusion via solving the PDE

$$\frac{1}{2}\Delta\mu - f.\nabla\mu - div(f)\mu = 0,$$

from random sample of μ.

[1] Maximum-likelihood estimation for diffusion processes via closed-form density expansions. AOS, 2013.

[2] Nonparametric statistical inference for drift vector fields of multi-dimensional diffusions. AOS, 2020.

[3] Semiparametric estimation of McKean-Vlasov SDEs, arXiv:2107.00539.

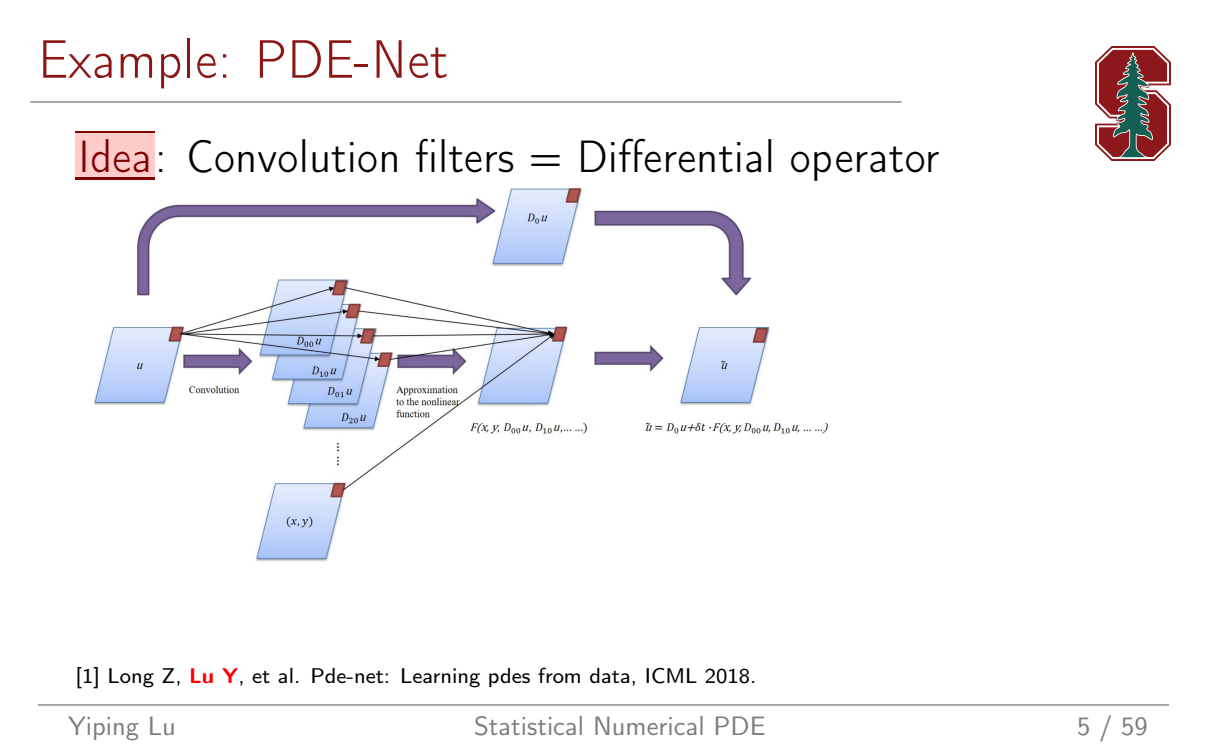
Example: High Dimensional PDE

High Dimensional PDE is hard to solve.

- Han J, Jentzen A, Weinan E. Solving high-dimensional partial differential equations using deep learning. Proceedings of the National Academy of Sciences, 2018, 115(34): 8505-8510.
- Han J, Hu R. Deep fictitious play for finding Markovian Nash equilibrium in multi-agent games Mathematical and Scientific Machine Learning. PMLR, 2020: 221-245.
- Shi X, Xu D, Zhang Z. Deep Learning Algorithms for Hedging with Frictions. arXiv preprint arXiv:2111.01931, 2021.

Potential Application: High dimensional control problem.

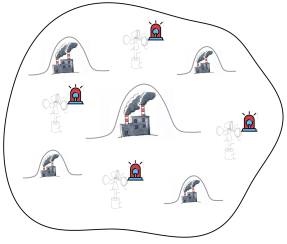
Example: PDE-Net

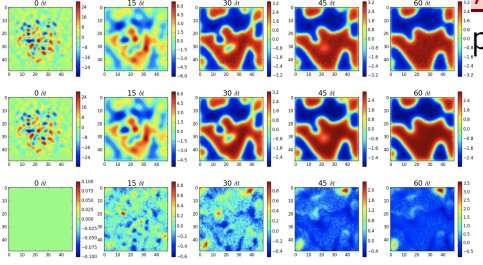


Idea: Convolution filters = Differential operator

[1] Long Z, **Lu Y**, et al. Pde-net: Learning pdes from data, ICML 2018.

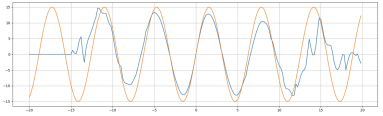
Example: PDE-Net





$$\frac{\partial u}{\partial t} = c\Delta u + \underbrace{f_s(u)}_{\text{unknown source}}$$

Aim: Predict the dynamic and source of population at the same time.



Example: PDE-Net

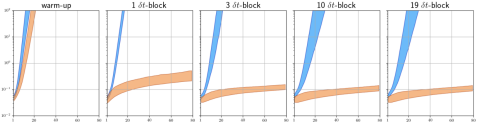
Question: Can we learn the filters?

Idea: Differential cancels low order polynomials

Momentum Condition

For all $|\beta| < |\alpha|$ and $|\beta| = |\alpha|$ but $\beta \neq \alpha$

$$\sum_{k \in \mathbb{Z}^2} k^\beta q[k] = 0$$



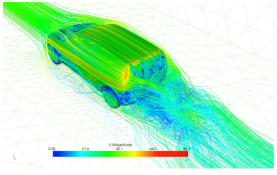
Blue: Frozen, Orange: Learn Filters

Learning filters gives a stable numerical scheme.

[1] Long Z, **Lu Y**, et al. Pde-net: Learning pdes from data, ICML 2018.

Motivation

How much information the physics can tells us?



$$\nabla \cdot \vec{u} = 0$$

$$\rho\left(\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u}\right) = -\nabla p + \mu \nabla^2 \vec{u} + \rho \vec{F}$$

Observational Data ⟷ **Physical Knowledge**

Questions Aim to Answer

Satistical Limit. For a given PDE , how large the sample size are needed to reach a prescribed performance level?

Optimal Estimators. How complex the model are needed to reach the satistical limit?

Computational Power. How can we design an algorithm?

Questions Aim to Answer

Satistical Limit. For a given PDE , how large the sample size are needed to reach a prescribed performance level?

Optimal Estimators. How complex the model are needed to reach the satistical limit?

Computational Power. How can we design an algorithm?

Questions Aim to Answer

Satistical Limit. For a given PDE , how large the sample size are needed to reach a prescribed performance level?

Optimal Estimators. How complex the model are needed to reach the satistical limit?

Computational Power. How can we design an algorithm?

1. Problem Formulation

2. Lower Bound

3. Upper Bound
   Empirical Risk Minimization
   Gradient Descent

Problem Formulation

Problem Formulation

Static Schrödinger Equation

$$\begin{aligned} -\Delta u + Vu &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{1}$$

What we observed:

- Random Samples in Domain: $\{x_i\}_{i=1}^n \sim \text{Unif}(\Omega)$
- RHS Function Values: $\{f_i = f(x_i) + \eta_i\}_{i=1}^n$

What we want:

- An Esitmate of u in **Sobolev Norm**.

Problem Formulation

Strong form (residual minimization) → Physics Informed Neural Network/DGM

$$\mathcal{L}(u) := |(-\Delta + V)u - f|^2_{L^2(\Omega)}$$

Variational form → Deep Ritz Methods

$$u^* = \arg \min_{u \in H^1(\Omega)} \mathcal{E}(u) := \frac{1}{2} \int_\Omega \|\nabla u\|^2 + V\|u\|^2 u(x) - \int_\Omega f u(x)$$

Problem Formulation

Strong form (residual minimization) → Physics Informed Neural Network/DGM

$$\mathcal{L}(u) := |(-\Delta + V)u - f|^2_{L^2(\Omega)}$$

Variational form → Deep Ritz Methods

$$u^* = \arg\min_{u \in H^1(\Omega)} \mathcal{E}(u) := \frac{1}{2}\int_\Omega \|\nabla u\|^2 + V\|u\|^2 u(x) - \int_\Omega f u(x)$$

Lower Bound

Lower Bound

Information Theoretical Lower Bound

Any Estimator H using $(X_i, f_i)_{i=1}^n$ can't do better than

$$\inf_{H} \sup_{u \in C^{\alpha}(\Omega)} \mathbb{E} \| H(\{X_i, f_i\}_{i=1,\cdots,n}) - u^* \|_{W_1^2} \gtrsim n^{-\frac{2\alpha-2}{2\alpha-4+d}},$$

- Solution $u \in H^{\alpha}$
- Consider Convergence in H^1 norm.

Lower Bound

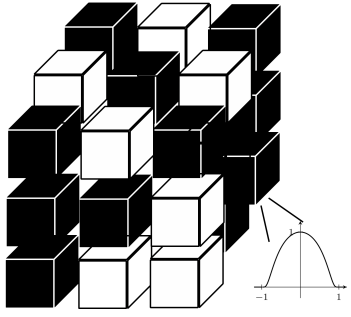
(X_i, f_i) doesn't have enough information of the PDE soultion, *i.e.* We can have two Different PDEs but generate Similar (X_i, f_i).

Turning Solving a PDE to Testing problems

- Construct several PDEs whose solution is sperated by Δ
- Give you a dataset, can you test which PDE generated data?
- Using Fano's method to have the lower bound.

Hypothesis used in Lower Bound

$$u_k(x) = \sum_{j \in [m]^d} \frac{\tau_j^{(k)}}{m^{s+\frac{d}{2}}} g(m(x - x^{(j)})), k = 1, \cdots, 2^{m^d/8},$$



- τ^k: selection of cubes, can be selected as a packing
- g: add a bump function
- $m^{s+\frac{d}{2}}$: statisfies $u \in H^s$

Hypothesis used in Lower Bound

$$u_k(x) = \sum_{j \in [m]^d} \frac{\tau_j^{(k)}}{m^{s+\frac{d}{2}}} g(m(x - x^{(j)})), k = 1, \cdots, 2^{m^d/8},$$

- Similarity of the Data: The data is sampled according to $f = \Delta u + Vu$. The similarity is ℓ_2 **Norm** of f.
- Difference of the PDEs: **Sobolev Norm** of u.

Every time you take a gradient, a m will comes out.

General Lower Bound

Information Theoretical Lower Bound

Any Estimator H using $(X_i, f_i)_{i=1}^n$ can't do better than

$$\inf_{H} \sup_{u \in C^{\alpha}(\Omega)} \mathbb{E} \| H(\{X_i, f_i\}_{i=1,\cdots,n}) - u^* \|_{W_s^2} \gtrsim n^{-\frac{2\alpha-2s}{2\alpha-2t+d}},$$

For

- t-th order PDE
- Solution $u \in H^{\alpha}$
- Consider Convergence in H^s

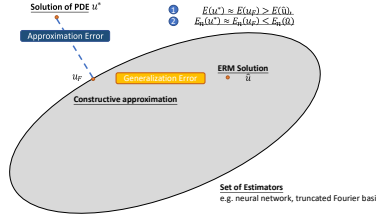
Now:
PINN: H^2 norm
DRM: H^1 norm

Upper Bound

Error Decomposition

If we

$$\mathbb{E}\left(\mathcal{E}(u_n) - \mathcal{E}(u^*)\right) \leqslant \underbrace{\mathbb{E}[\mathcal{E}(u_n) - \mathcal{E}_n(u_n)]}_{\Delta\mathcal{E}_{\mathbf{gen}}} + \underbrace{\mathbb{E}[\mathcal{E}_n(u_{\mathcal{F}})] - \mathcal{E}(u_{\mathcal{F}})}_{\Delta\mathcal{E}_{\mathbf{bias}}} + \underbrace{\mathcal{E}(u_{\mathcal{F}}) - \mathcal{E}(u^*)}_{\Delta\mathcal{E}_{\mathbf{approx}}}.$$



bias+variance decomposition:

$$\text{approximation} + \frac{\text{Complexity}}{\sqrt{n}} \text{ bound}$$

But leads to sub-optimal results... [Shin et al 2020], [Lu et al 2021], [Duan et al 2021]

Motivating Example

Estimating the mean

Goal. Estimate $\theta = \mathbb{E}[X]$ via loss function $\frac{1}{2}(\theta - x)^2$

Empirical Solution of ℓ_2 loss: $\theta_n = \frac{1}{n}\sum_{i=1}^n x_i$, using chernoff bound we know $\theta_n - \theta = \sqrt{\frac{\sigma^2 \log \frac{1}{\delta}}{n}}$ w.h.p.

The generalization gap $L(\theta_n) - L(\theta^*) = \|\theta - \theta^*\|^2$ w.h.p

$$L(\theta_n) - L(\theta^*) = (\theta_n - \theta^*)^2 \leqslant C\frac{\sigma^2 \log \frac{1}{\delta}}{n}$$

A $O(\frac{1}{n})$ fast rate bound.

Motivating Example

Estimating the mean

Goal. Estimate $\theta = \mathbb{E}[X]$ via loss function $\frac{1}{2}(\theta - x)^2$

Empirical Solution of ℓ_2 loss: $\theta_n = \frac{1}{n}\sum_{i=1}^n x_i$, using chernoff bound we know $\theta_n - \theta = \sqrt{\frac{\sigma^2 \log \frac{1}{\delta}}{n}}$ w.h.p.

The generalization gap $L(\theta_n) - L(\theta^*) = \|\theta - \theta^*\|^2$ w.h.p

$$L(\theta_n) - L(\theta^*) = (\theta_n - \theta^*)^2 \leqslant C\frac{\sigma^2 \log \frac{1}{\delta}}{n}$$

A $O(\frac{1}{n})$ fast rate bound.

Observation 1: Fast rate via Localization

The variational form has some "strongly convex"

Lemma

Assume $0 < V_{\min} \leqslant V(x) \leqslant V_{\max}$ *for all* $x \in \Omega$

$$\frac{2}{\max(1, V_{\max})}\big(\mathcal{E}(u) - \mathcal{E}(u^*)\big) \leqslant \|u - u^*\|^2_{H^1(\Omega)} \leqslant \frac{2}{\max(1, V_{\min})}\big(\mathcal{E}(u) - \mathcal{E}(u^*)\big)$$

Can we have a $\frac{1}{n}$ fast rate generalization bound?

Relevant Work

non-parametric function estimation

- Holder spaces : [Schmidt-Hieber et al. 2020]
- Besov spaces and "mixed smotth" Besov spaces: [Suzuki et al. 2019]
- Low dimensional strucutre: [Suzuki et al 2019] [Chen et al. 2020]
- ...

Leads to $N^{-\frac{s}{d}+\frac{N}{n}}$, where N is the log size of the neural network.

Recall: Uniform Law via Rademacher Complexity

Recall that the Rademacher complexity of a function class $\mathcal{G}$ is defined by

$$R_n(\mathcal{G}) = \mathbb{E}_Z \mathbb{E}_\sigma \Big[\sup_{g \in \mathcal{G}} \Big| \frac{1}{n} \sum_{j=1}^{n} \sigma_j g(Z_j) \Big| \; \Big| \; Z_1, \cdots, Z_n \Big].$$

Symmetrization Lemma

Let $\mathcal{F}$ be a set of functions. Then $\mathbb{E} \sup_{u \in \mathcal{F}} \Big| \frac{1}{n} \sum_{j=1}^{n} u(X_j) - \mathbb{E}_{X \sim P_\Omega} u(X) \Big| \leqslant 2R_n(\mathcal{F})$.

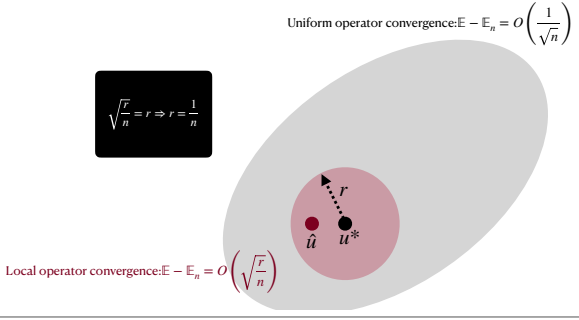
This Gives us $\sqrt{1/n}$ rate.

Local Rademacher Complexity

Local Rademacher Complexity

$$\psi(r) \geqslant \mathbb{E} R_n\{f \in \mathcal{F}, T(f) \leqslant r\}$$

The generalization bound: fix point solution of $\psi(r) = r$



Localization

You can have other ways to do localization

1. Easier way: Chapter 3 of `https://sites.stat.washington.edu/people/jaw/RESEARCH/TALKS/Delft/emp-proc-delft-big.pdf`
2. `https://www.theses.fr/2020IPPAG002.pdf` to regularization, robust learning, interpolation
3. Bartlett P L, Bousquet O, Mendelson S. Local rademacher complexities[J]. The Annals of Statistics, 2005, 33(4): 1497-1537.
4. For Gaussian Noise: `http://ibis.t.u-tokyo.ac.jp/suzuki/paper/note_ERM_generror.pdf`
5. Xu Y, Zeevi A. Towards Optimal Problem Dependent Generalization Error Bounds in Statistical Learning Theory. arXiv preprint arXiv:2011.06186, 2020

Be careful of the hidden assumptions.

Is Fast Rate Optimal?

For PINN, Yes!. For DRM, No!

Upper Bounds			**Lower Bound**
Objective Function	Neural Network	Fourier Basis	
Deep Ritz	$n^{-\frac{2s-2}{d+2s-2}} \log n$	$n^{-\frac{2s-2}{d+2s-2}}$	$n^{-\frac{2s-2}{d+2s-4}}$
PINN	$n^{-\frac{2s-4}{d+2s-4}} \log n$	$n^{-\frac{2s-4}{d+2s-4}}$	$n^{-\frac{2s-4}{d+2s-4}}$

Table: Upper bounds and lower bounds Fast Rate achieved.

Why?

Let's use kernel as an example

- Kernel is easy, we can know whether our bound is tight.
- Kernel is useful

Chen Y, Hosseini B, Owhadi H, et al. Solving and learning nonlinear PDEs with gaussian processes. arXiv preprint arXiv:2103.12959, 2021.

Richter L, Sallandt L, Nüsken N. Solving high-dimensional parabolic PDEs using the tensor train format. arXiv preprint arXiv:2102.11830, 2021.

Chen Y, Hoskins J, Khoo Y, et al. Committor functions via tensor networks. arXiv preprint arXiv:2106.12515, 2021.

A Fourier Basis View

Solving a simple PDE $\Delta u = f$ using Fourier Basis.

Estimator 1

First Estimate f then solve u, $f_z = \frac{1}{n}\sum f(x_i)\phi_z(x_i)$, then $u = \sum \frac{1}{\|z\|^2} f_z \phi_z(x)$

Estimator 2

Plug $u = \sum u_z \phi_z(x)$ into the Deep Ritz Objecive function

$$\frac{1}{n}\sum_{i=1}^{n}\left(\sum_z u_z \nabla\phi_z(x_i)\right)^2 + \sum_z u_z \phi_z(x_i) f(x_i)$$

A Fourier Basis View

Solving a simple PDE $\Delta u = f$ using Fourier Basis.

Estimator 1

First Estimate f then solve u, $f_z = \frac{1}{n}\sum f(x_i)\phi_z(x_i)$, then $u = \sum \frac{1}{\|z\|^2} f_z \phi_z(x)$

Estimator 2

Plug $u = \sum u_z \phi_z(x)$ into the Deep Ritz Objecive function

$$\frac{1}{n}\sum_{i=1}^{n}\left(\sum_z u_z \nabla \phi_z(x_i)\right)^2 + \sum_z u_z \phi_z(x_i) f(x_i)$$

A Fourier Basis View

Solving a simple PDE $\Delta u = f$ using Fourier Basis.

Estimator 1

First Estimate f then solve u, $f_z = \frac{1}{n}\sum f(x_i)\phi_z(x_i)$, then $u = \sum \frac{1}{\|z\|^2} f_z \phi_z(x)$

Estimator 2

Plug $u = \sum u_z \phi_z(x)$ into the Deep Ritz Objecive function

$$\frac{1}{n}\sum_{i=1}^{n}\left(\sum_z u_z \nabla\phi_z(x_i)\right)^2 + \sum_z u_z \phi_z(x_i) f(x_i)$$

Estimator1 is Optimal

Consider estimating in H_{-1} norm using Fourier Basis up to Z, *i.e.* $\mathcal{Z} := \{z \in \mathbb{N}^d | \|z\|_\infty \leqslant Z\}$.

- **Bias**:

$$\| \sum_{\|z\|_\infty > Z} f_z \phi_z \|_{H^{-1}}^2 \leqslant C \sum_{\|z\|_\infty > Z} f_z^2 z^{-2} \leqslant \|z\|^{-2(s-1)} \|f\|_{H_{\alpha-2}}^2$$

- **Variance**:

$$\mathbb{E}\|f - f\|_{H_{-1}}^2 \leqslant \mathbb{E} \sum_{\|z\|_\infty \leqslant Z} (f_z - f_z)^2 \|\phi_z\|_{H_{-1}}^2 \leqslant \sum_{\|z\|_\infty \leqslant Z} |z|^{-1} \mathrm{Var}(f_z)$$

Final bound: $Z^{-2(s-1)} + \frac{Z^{d-2}}{n}$

Difference Between Estimator1 and 2

- **Estimator 1:** The Fourier coefficient of the solution of Estimator 1 is

$$\mathbf{u}_{1,z} = \operatorname{diag}\left(\|z\|_2^2\right)^{-1}_{\|z\|_\infty \leqslant Z} f_z. \tag{2}$$

- **Estimator 2:** The Fourier coefficient of the solution of Estimator 2 is

$$\mathbf{u}_{2,z} = \underbrace{\left(\frac{1}{n}\sum_{i=1}^{n} \nabla\phi_i(x_i)\nabla\phi_j(x_i)\right)^{-1}_{\|i\|_\infty \leqslant Z, \|j\|_\infty \leqslant Z}}_{\text{empirical Gram Matrix } A} f_z, \tag{3}$$

Thus $\|u_1 - u_2\|_{H_1}^2 \propto \|((\mathbb{E}A) - A)\|_{\mathrm{H}}^2 \propto \frac{Z^d}{n}$

Difference Between Estimator1 and 2

- **Estimator 1:** The Fourier coefficient of the solution of Estimator 1 is

$$\mathbf{u}_{1,z} = \operatorname{diag}\left(\|z\|_2^2\right)^{-1}_{\|z\|_\infty \leqslant Z} f_z. \tag{2}$$

- **Estimator 2:** The Fourier coefficient of the solution of Estimator 2 is

$$\mathbf{u}_{2,z} = \underbrace{\left(\frac{1}{n}\sum_{i=1}^{n} \nabla\phi_i(x_i)\nabla\phi_j(x_i)\right)^{-1}_{\|i\|_\infty \leqslant Z, \|j\|_\infty \leqslant Z}}_{\text{empirical Gram Matrix } A} f_z, \tag{3}$$

Thus $\boxed{\|u_1 - u_2\|_{H_1}^2 \propto \|((\mathbb{E}A) - A)\|_{\mathrm{H}}^2 \propto \frac{Z^d}{n}}$.

How Much Gradient We Need?

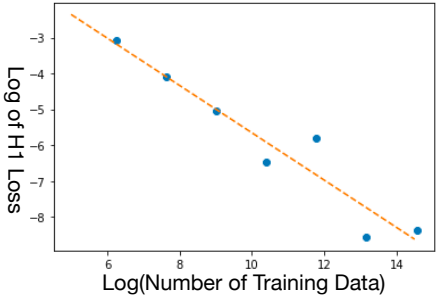
We Introduce the Modified DRM

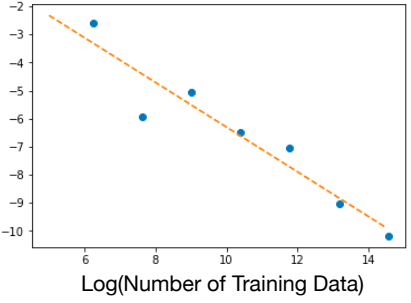
$$\mathcal{E}_{N,n}^{\mathrm{MDRM}}(u) = \underbrace{\frac{1}{N}\sum_{j=1}^{N}\left[|\Omega|\cdot\frac{1}{2}\|\nabla u(X_j')\|^2\right]}_{\text{Sample More Gradients}} + \frac{1}{n}\sum_{j=1}^{n}\left[|\Omega|\cdot\left(\frac{1}{2}V(X_j)|u(X_j)|^2 - f_j u(X_j)\right)\right] \tag{4}$$

Thus Variance: $\frac{\xi^d}{N} < \frac{\xi^{d-2}}{n} \simeq \xi^{-2(s-1)} \Rightarrow \xi \simeq n^{\frac{1}{d+2s-4}}$ and

$$\frac{N}{n} = \xi^2 = n^{\frac{2}{d+2s-4}}$$

Experiment





	(a) Deep Ritz Methods	(b) Modified Deep Ritz Methods
Theory	$\frac{2s-2}{d+2s-2} = 0.75$	$\frac{2s-2}{d+2s-4} = 1$
Empirical	0.6595	0.7953
R2 Score	0.91	0.89

Summarize in One Table...

Upper Bounds			**Lower Bound**
Objective Function	Neural Network	Fourier Basis	
Deep Ritz	$n^{-\frac{2s-2}{d+2s-2}} \log n$	$n^{-\frac{2s-2}{d+2s-2}}$	$n^{-\frac{2s-2}{d+2s-4}}$
Modified Deep Ritz	$n^{-\frac{2s-2}{d+2s-2}} \log n$	$n^{-\frac{2s-2}{d+2s-4}}$	$n^{-\frac{2s-2}{d+2s-4}}$
PINN	$n^{-\frac{2s-4}{d+2s-4}} \log n$	$n^{-\frac{2s-4}{d+2s-4}}$	$n^{-\frac{2s-4}{d+2s-4}}$

Table: Upper bounds and lower bounds we achieve in this paper and previous work. The upper bound colored in red indicates that the convergence rate matches the min-max lower bound.

Observation 3: Tigher Local Rademacher

Local Rademacher Complexity

$$\psi(r) \geqslant \mathbb{E}R_n\{f \in \mathcal{F}, \underbrace{T(f) \leqslant r}_{\text{loss function}}\}$$

- For nonparametric estimation: ℓ_2 Norm
- For Solving PDE: Sobolev Norm

Can Tigher Norm leads to Tigher Bound?

- Fourier Basis Yes DNN No

Recall: Estimator1 is Optimal

Consider estimating in H_{-1} norm using Fourier Basis up to Z, *i.e.* $\mathcal{Z} := \{z \in \mathbb{N}^d | \|z\|_\infty \leqslant Z\}$.

- **Bias**:

$$\| \sum_{\|z\|_\infty > Z} f_z \phi_z \|_{H^{-1}}^2 \leqslant C \sum_{\|z\|_\infty > Z} f_z^2 z^{-2} \leqslant \|z\|^{-2(s-1)} \|f\|_{H_{\alpha-2}}^2$$

- **Variance**:

$$\mathbb{E}\|f - f\|_{H_{-1}}^2 \leqslant \mathbb{E} \sum_{\|z\|_\infty \leqslant Z} (f_z - f_z)^2 \|\phi_z\|_{H_{-1}}^2 \leqslant \sum_{\|z\|_\infty \leqslant Z} |z|^{-1} \mathrm{Var}(f_z)$$

Final bound: $Z^{-2(s-1)} + \frac{Z^{d-2}}{n}$ (Not number of Basis $\frac{Z^d}{n}$)

Tigher Local Rademacher

$\|f\|^2_{H^1(\Omega)} \leqslant \rho$ implies constraint $\sum_{\|z\|_\infty \leqslant \xi} |f_z|^2 \|z\|^2 \lesssim \rho$ on the Fourier coefficients

For the Rademacher complexity can be bounded by

$$\frac{1}{n}\sum_{i=1}^{n} \sigma_i f(X_i) \leqslant \frac{1}{n}\Big(\sum_{\|z\|_\infty \leqslant \xi} |f_z|^2 \|z\|^2 \Big)^{\frac{1}{2}} \Big(\sum_{\|z\|_\infty \leqslant \xi} \Big| \sum_{i=1}^{n} \frac{\sigma_i}{\|z\|} \phi_z(X_i) \Big|^2 \Big)^{\frac{1}{2}}$$

You can have an improvement from $\frac{1}{\sqrt{n}} \xi^d$ to $\frac{1}{\sqrt{n}} \xi^{d-2}$,

But the neural network estimator the local Rademacher complexity is always $\sqrt{\frac{N}{n}}$

Tigher Local Rademacher

$\|f\|^2_{H^1(\Omega)} \leqslant \rho$ implies constraint $\sum_{\|z\|_\infty \leqslant \xi} |f_z|^2 \|z\|^2 \lesssim \rho$ on the Fourier coefficients

For the Rademacher complexity can be bounded by

$$\frac{1}{n}\sum_{i=1}^{n} \sigma_i f(X_i) \leqslant \frac{1}{n}\Big(\sum_{\|z\|_\infty \leqslant \xi} |f_z|^2 \|z\|^2 \Big)^{\frac{1}{2}} \Big(\sum_{\|z\|_\infty \leqslant \xi} \Big| \sum_{i=1}^{n} \frac{\sigma_i}{\|z\|} \phi_z(X_i) \Big|^2 \Big)^{\frac{1}{2}}$$

You can have an improvement from $\frac{1}{\sqrt{n}}\xi^d$ to $\frac{1}{\sqrt{n}}\xi^{d-2}$,

But the neural network estimator the local Rademacher complexity is always $\sqrt{\frac{N}{n}}$

Gradient Descent

Why you select Ritz form in the first paper

Me

minimizing $\int(\Delta u)^2$ is crazy to me due to the condition number of $\Delta^\top\Delta$

Lexing

Gradient Descent

Why you select Ritz form in the first paper

Me

minimizing $\int(\Delta u)^2$ is crazy to me due to the condition number of $\Delta^\top\Delta$

Lexing

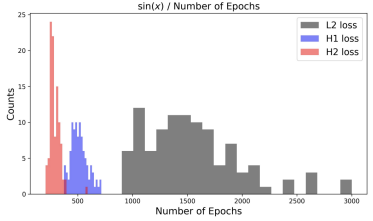
Gradient Descent

Why you select Ritz form in the first paper

Me

minimizing $\int(\Delta u)^2$ is crazy to me due to the condition number of $\Delta^\top \Delta$

Lexing



(Stochastic) Gradient Descent

Let's consider $\Delta u = f$ via minimizing $\frac{1}{2}\langle f, \mathcal{A}_1 f\rangle - \langle u, \mathcal{A}_2 f\rangle$

- **Deep Ritz Methods**. $\mathcal{A}_1 = \Delta, \mathcal{A}_1 = Id$
- **PINN**. $\mathcal{A}_1 = \Delta^2, \mathcal{A}_1 = \Delta$

We consider parameterize f using kernel regression $f(x) = \langle \theta, K_x\rangle$.
Then we apply a stochastic gradient descent and get

$$\theta_{t+1} = \theta_t - \eta(\langle \theta, \mathcal{A}_1 K_{x_i}\rangle K_{x_i} - f_i \mathcal{A}_2 K_{x_i})$$

(Stochastic) Gradient Descent

Let's consider $\Delta u = f$ via minimizing $\frac{1}{2}\langle f, \mathcal{A}_1 f\rangle - \langle u, \mathcal{A}_2 f\rangle$

- **Deep Ritz Methods**. $\mathcal{A}_1 = \Delta, \mathcal{A}_1 = Id$
- **PINN**. $\mathcal{A}_1 = \Delta^2, \mathcal{A}_1 = \Delta$

We consider parameterize f using kernel regression $f(x) = \langle \theta, K_x\rangle$.
Then we apply a stochastic gradient descent and get

$$\theta_{t+1} = \theta_t - \eta(\langle \theta, \mathcal{A}_1 K_{x_i}\rangle K_{x_i} - f_i \mathcal{A}_2 K_{x_i})$$

Assumptions On Kernel

Besides the common assumptions for Kernel

- **Capacity Condition.** The population covariance Σ matrix have fast eigen decay $\mathrm{tr}[(\mathbb{E}K_x \otimes K_x)^{1/\alpha}] < \infty$
- **Source Condition.** The target function is smooth $f_* = \mathcal{L}^r \phi, \phi \in \mathcal{H}$, *i.e.* $\|\Sigma^{1/2-\beta}\theta_*\| < \infty$.
- **Regularity Condition.** $\|g\|_{L_\infty} \lesssim \|\Sigma^{\frac{1-\mu}{2}} g\|_H$

Further assumes

$\mathcal{A}_i$ eigen decay (p, q) fast and is commutable with Σ.

Holds for Shift invariant kernel on torus and NTK on sphere.

Assumptions On Kernel

Besides the common assumptions for Kernel

- **Capacity Condition.** The population covariance Σ matrix have fast eigen decay $\mathrm{tr}[(\mathbb{E}K_x \otimes K_x)^{1/\alpha}] < \infty$
- **Source Condition.** The target function is smooth $f_* = \mathcal{L}^r \phi, \phi \in \mathcal{H}$, *i.e.* $\|\Sigma^{1/2-\beta}\theta_*\| < \infty$.
- **Regularity Condition.** $\|g\|_{L_\infty} \lesssim \|\Sigma^{\frac{1-\mu}{2}} g\|_H$

Further assumes

$\mathcal{A}_i$ eigen decay (p, q) fast and is commutable with Σ.

Holds for Shift invariant kernel on torus and NTK on sphere.

Assumptions On Kernel

Besides the common assumptions for Kernel

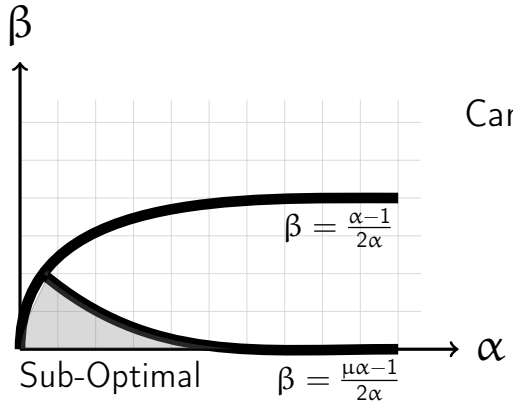
- **Capacity Condition.** The population covariance Σ matrix have fast eigen decay $\mathrm{tr}[(\mathbb{E}K_x \otimes K_x)^{1/\alpha}] < \infty$
- **Source Condition.** The target function is smooth $f_* = \mathcal{L}^r \phi, \phi \in \mathcal{H}$, *i.e.* $\|\Sigma^{1/2-\beta}\theta_*\| < \infty$.
- **Regularity Condition.** $\|g\|_{L_\infty} \lesssim \|\Sigma^{\frac{1-\mu}{2}} g\|_H$

Further assumes

$\mathcal{A}_i$ eigen decay (p, q) fast and is commutable with Σ.

Holds for Shift invariant kernel on torus and NTK on sphere.

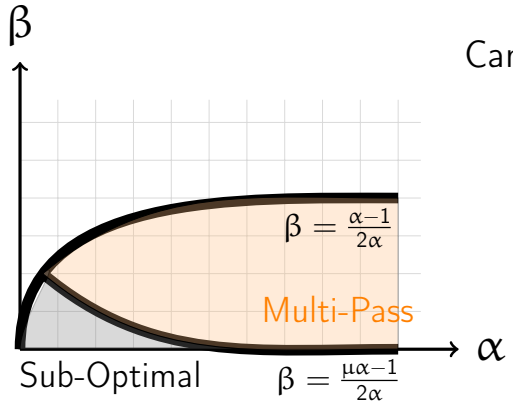
Review: (S)GD for Kernel Regression



Can be concluded into Three Regimes

- $2\alpha\beta + 1 \geqslant \mu\alpha$: Suboptimal, concentration error of $\frac{1}{n}K_x \otimes K_x \to \Sigma$ dominates

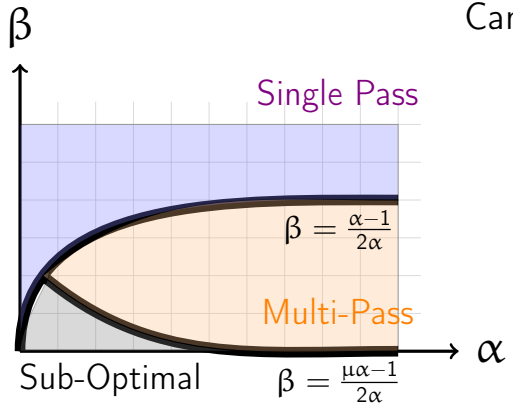
Review: (S)GD for Kernel Regression



Can be concluded into Three Regimes

- $2\alpha\beta + 1 \geqslant \mu\alpha$: Suboptimal, concentration error of $\frac{1}{n}K_x \otimes K_x \to \Sigma$ dominates
- $\mu\alpha < 2\alpha\beta + 1 < \alpha$: Constant Lr, Multipass

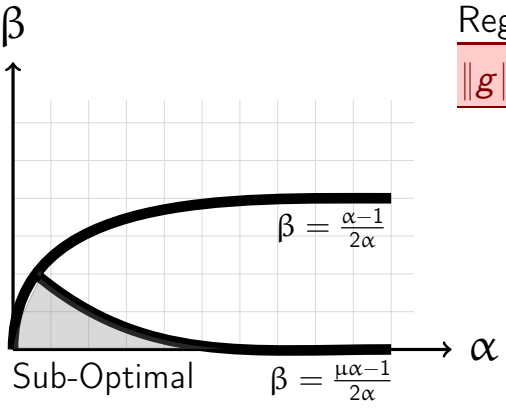
Review: (S)GD for Kernel Regression



Can be concluded into Three Regimes

- $2\alpha\beta + 1 \geqslant \mu\alpha$: Suboptimal, concentration error of $\frac{1}{n}K_x \otimes K_x \to \Sigma$ dominates
- $\mu\alpha < 2\alpha\beta + 1 < \alpha$: Constant Lr, Multipass
- $2\alpha r + 1 > \alpha$: Small Lr, Single Pass

Sub-Optimal Regime



Sub Optimal Regime relates to the Regularity Condition of the Kernel

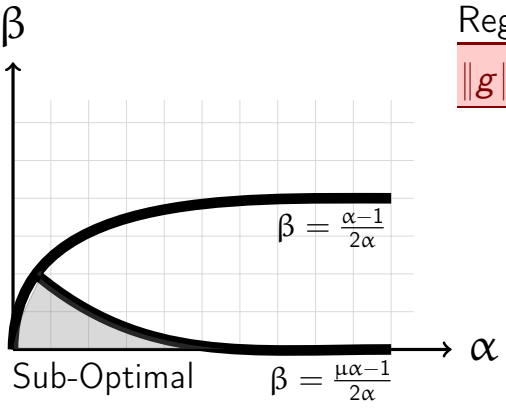
$$\|g\|_{L_\infty} \lesssim \|\Sigma^{\frac{1-\mu}{2}} g\|_H,$$

- Sub-Optimal due to the concentration of

$$1/n\sum_{i=1}^{n} K_{x_i} \otimes K_{x_i} \to \mathbb{E}[K_{x_i} \otimes K_{x_i}]$$

- Can be reduced using Semi-supervised Learning [Murata & Suzuki 2021]

Sub-Optimal Regime



Sub Optimal Regime relates to the Regularity Condition of the Kernel

$\|g\|_{L_\infty} \lesssim \|\Sigma^{\frac{1-\mu}{2}} g\|_H$,

- Sub-Optimal due to the concentration of

  $1/n\sum_{i=1}^n K_{x_i} \otimes K_{x_i} \to \mathbb{E}[K_{x_i} \otimes K_{x_i}]$

- Can be reduced using Semi-supervised Learning [Murata & Suzuki 2021]

Sobolev Learning Rate: Setting

We can formulate the Sobolev Norm as $[H^\alpha]$ norm as

$$\left\| \sum_{i \geqslant 1} a_i \mu_i^{\alpha/2} e_i \right\|_{[H]^\alpha} := \left(\sum_{i \geqslant 1} a_i^2 \right)^2$$

- The evaulation Sobolev norm can be different as the training Sobolev norm. We consider convergence rate in H^γ norm.
- The same as the Source Condition

$$\|f\|_{[H^\alpha]} \leqslant \|\Sigma^{\frac{1-\alpha}{2}} f\|_H$$

We want $n^{-\frac{(\beta-\gamma)\alpha}{\beta\alpha+2(p-q)+1}}$

Recall

$$\inf_{H} \sup_{u\in C^{\alpha}(\Omega)} \mathbb{E}\|H(\{X_i, f_i\}_{i=1,\cdots,n}) - u^*\|_{W_s^2} \gtrsim n^{-\frac{2\alpha-2s}{2\alpha-2t+d}},$$

and translate it into kernel setting

$$\|f_\lambda - f\|^2_{[H]^\gamma} \leqslant n^{-\frac{(\beta-\gamma)\alpha}{\beta\alpha+2(p-q)+1}}$$

They matches for

- $\alpha = 1/d$
- $\beta = 2\alpha$, $\gamma = 2s$
- $(q-p) = t$

Main Results

We can achieve infomration theortical optimal rate $n^{-\frac{(\beta-\gamma)\alpha}{\beta\alpha+2(p-q)+1}}$ via

- Bias $\lambda^{\frac{(\beta-\gamma)\alpha}{\alpha+p}}$
- Variance $\frac{1}{n}$ $\underbrace{\lambda^{-\frac{\gamma\alpha+p}{\alpha+p}}}_{\|\Sigma^{\frac{1-\gamma}{2}}\Sigma_{\mathcal{A}_1}\|}$ $\underbrace{\lambda^{-\frac{1}{\alpha+p}}\lambda^{-\frac{p-2q}{\alpha+p}}}$ Effective Dim: $\mathrm{tr}((\Sigma_{\mathcal{A}_1}+\lambda)^{-1}\Sigma_{\mathcal{A}_2^\top\mathcal{A}_2})$
- Convergence of Covariance $\frac{1}{n}\lambda^{-\frac{\mu\alpha-p}{\alpha+p}}\|\Sigma^{\frac{1-\gamma}{2}}\Sigma_{\mathcal{A}_1}\|\|\mathcal{A}_2(f^*-f_\lambda^*)\|_{L_2}^2 = \frac{1}{n}\|\Sigma^{\frac{1-\gamma}{2}}\Sigma_{\mathcal{A}_1}\|\lambda^{-\frac{\mu\alpha-p}{\alpha+p}}\lambda^{\frac{\beta\alpha-2q}{\alpha+p}}$

Convergence time

The convergence time will equal to the optimal selection of λ

Iteration Time

$$\lambda = n^{\frac{\alpha+p}{\beta\alpha+2(p-q)+1}}$$

- Independent of γ.
- β larger, the impact of p will smaller.

DRM Vs PINN

Recall Iteration time $\lambda = n^{\frac{\alpha+p}{\beta\alpha+2(p-q)+1}}$. To compare DRM and PINN, we should fix $p = q$ and then consider the dependency of iteration time on p.

- Denominator do nothing with p
- Numerator
  - $p < 0, \alpha > 0$, differential operator helps to balance the condition number of the kernel operator. PINN is faster.
  - $\alpha + p > 0$ means activation function should be smooth for NTK

DRM Vs PINN

Recall Iteration time $\lambda = n^{\frac{\alpha+p}{\beta\alpha+2(p-q)+1}}$. To compare DRM and PINN, we should fix $p = q$ and then consider the dependency of iteration time on p.

- Denominator do nothing with p
- Numerator
  - $p < 0, \alpha > 0$, differential operator helps to balance the condition number of the kernel operator. PINN is faster.
  - $\alpha + p > 0$ means activation function should be smooth for NTK

DRM Vs PINN

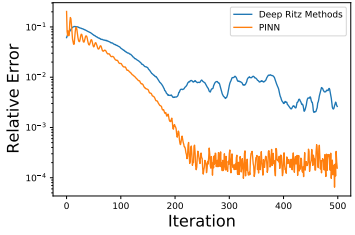


Figure: $\sum_{i=1}^{d} \sin(2\pi x)$

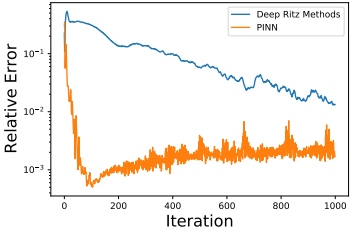


Figure: $\sum_{i=1}^{d} \sin(4\pi x)$

Variance of Integral by Parts

$$\mathbb{E}_{\mathbb{P}_n(x,y)} \frac{1}{2} \langle u, K_x \otimes \mathcal{A}_1 K_x u \rangle - y \langle u, \mathcal{A}_2 K_x \rangle$$

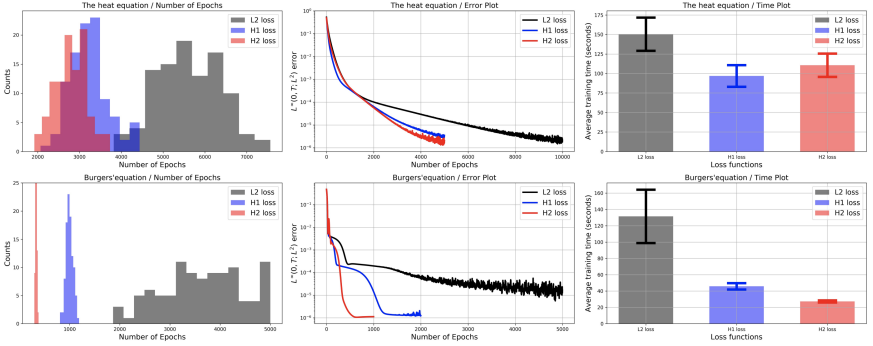
We considered the dynamic

$$\theta_t = \theta_{t-1} + \gamma \frac{1}{n} \sum_{i=1}^{n} \left(y_i \mathcal{A}_2 K_{x_i} - \underbrace{\langle \theta_{t-1}, \mathcal{A}_1 K_{x_i} \rangle_{\mathcal{H}} K_{x_i}}_{\text{not } \left(\langle \theta_{t-1}, \mathcal{A}_1 K_{x_i} \rangle_{\mathcal{H}} K_{x_i} + \langle \theta_{t-1}, K_{x_i} \rangle_{\mathcal{H}} \mathcal{A}_1 K_{x_i} \right)} \right)$$

for the variance of integral by parts may dominated.

Sobolev Training is Faster

How many gradients used for training? Matches α!

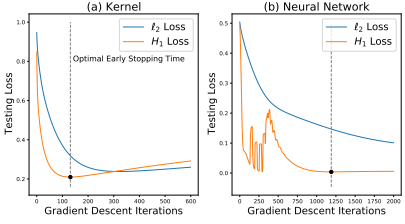


Sobolev training can also only depend on function vaule

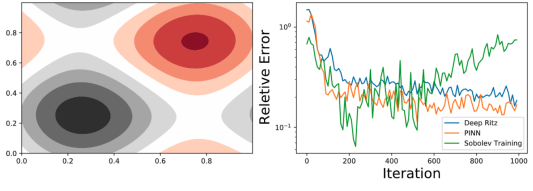
$$\int |\nabla u - \nabla f|^2 dx = \int \|\nabla u\|_2^2 + 2\Delta u \cdot f + \|\nabla f\|_2^2 dx$$

Sobolev Training is Faster

- Sobolev Estimating a function



- Sobolev Training a PDE



Furture Work

- optimal neural network based optimal Ritz method, it's even hard for two-layer case
- norm/rademecher complexity constrained approximation bounds
- theory for operator learning
- scale up the Sobolev training to high dimension
- stochastic gradient descent and online learning?
- the variance of integral by parts?

Take Home Message

- Non-parametric statistics view of numerical PDE solver
- Gives us new constraints to design objective functions to be statistical/information theortical optimal
- sparsity of the weight is not a good measurement of the complexity of gradients, we need to find new measure
- GD analysis sugguest Sobolev Training

Other Examples? Application in operation research?

Reference

- Long Z, Lu Y, Ma X, et al. Pde-net: Learning pdes from data, ICML 2018.
- Lu Y, Chen H, Lu J, et al. Machine Learning For Elliptic PDEs: Fast Rate Generalization Bound, Neural Scaling Law and Minimax Optimality. ICLR 2021.
- Lu Y, Jose B, et al. Sobolev Acceleration and Statistical Optimality for Learning Elliptic Equations, submitted.

Thank you for listening!
and Questions?

Yiping Lu

yplu@stanford.edu.cn
https://web.stanford.edu/~yplu/