Deep Learning Theory

Taiji Suzuki

The University of Tokyo

Deep Learning Theory Team/AIP-RIKEN

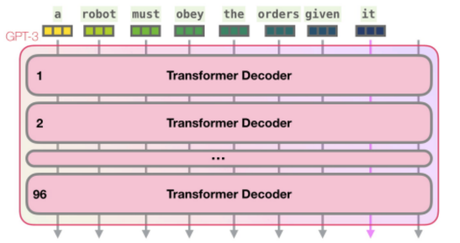
12nd/Mar/2024

MLSS2024@OIST

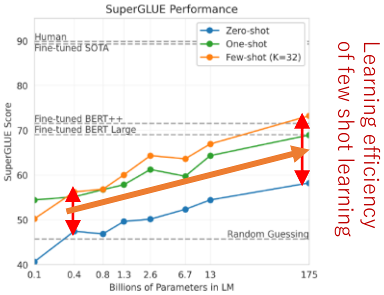
Success of deep learning

Deep learning has shown great performances in the AI research field.
→ Why?

Large language model



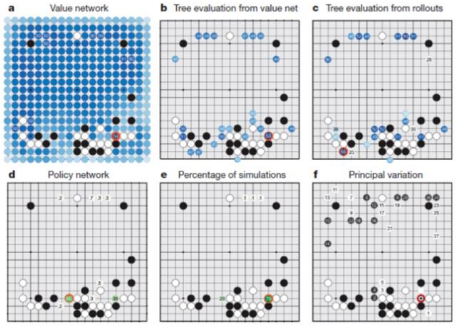
[Alammar: How GPT3 Works - Visualizations and Animations, https://jalammar.github.io/how-gpt3-works-visualizations-animations/]



Performance of few-shot learning against model size

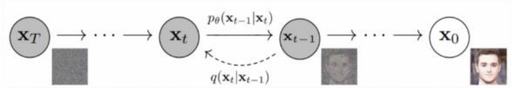
[Brown et al. “Language Models are Few-Shot Learners”, NeurIPS2020]

AlphaGo/Zero



[Silver et al. (Google Deep Mind): Mastering the game of Go with deep neural networks and tree search, Nature, 529, 484—489, 2016]

Generative models (diffusion models)



[Ho, Jain, Abbeel: Denoising Diffusion Probabilistic Models. 2020]

Image recognition

[He, Gkioxari, Dollár, Girshick: Mask R-CNN, ICCV2017]

Stable diffusion, 2022.

Jason Allen "Théâtre D'opéra Spatial“ generated by **Midjourney**.
Colorado State Fa
1st prize in digital

Generated by NovelAI

What we need to solve?

Why does deep learning work well?

- Several theoretical work has been conducted.
- There are still many things that should be explored.

- Clarification of principle of deep learning
- What is essential to realize a “good” learning system?

→**We may find a new method beyond DL**.

Issue in academic conference

Ali Rahimi's talk at NIPS(NIPS 2017 Test-of-time award presentation)

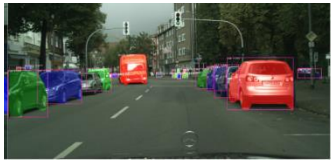
Ali Rahimi's talk at NIPS(NIPS 2017 Test-of-time award presentation)

Ali Rahimi’s talk at NIPS2017 (test of time award).
“Random features for large-scale kernel methods.”

Criticism that DL is “alchemy.”

Issue in industries

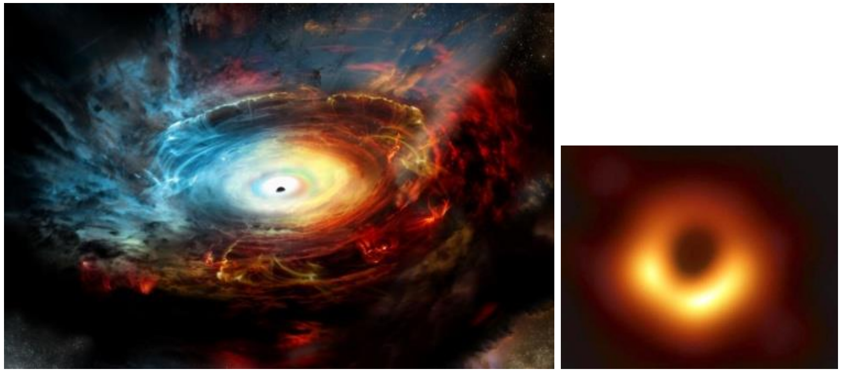
- We don’t want to use black-box system.
- Accountabilities of companies.



Deep learning theory

Role of mathematics

Physics



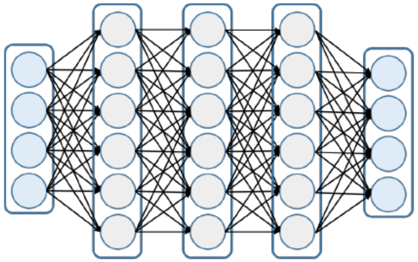
physical phenomenon

$$R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R = \frac{8\pi G}{c^4} T_{\mu\nu}$$

- Theory of Relative
  - ➢ Riemannian geometry
- Quantum mechanics
  - ➢ Functional analysis

Mathematics

Machine learning



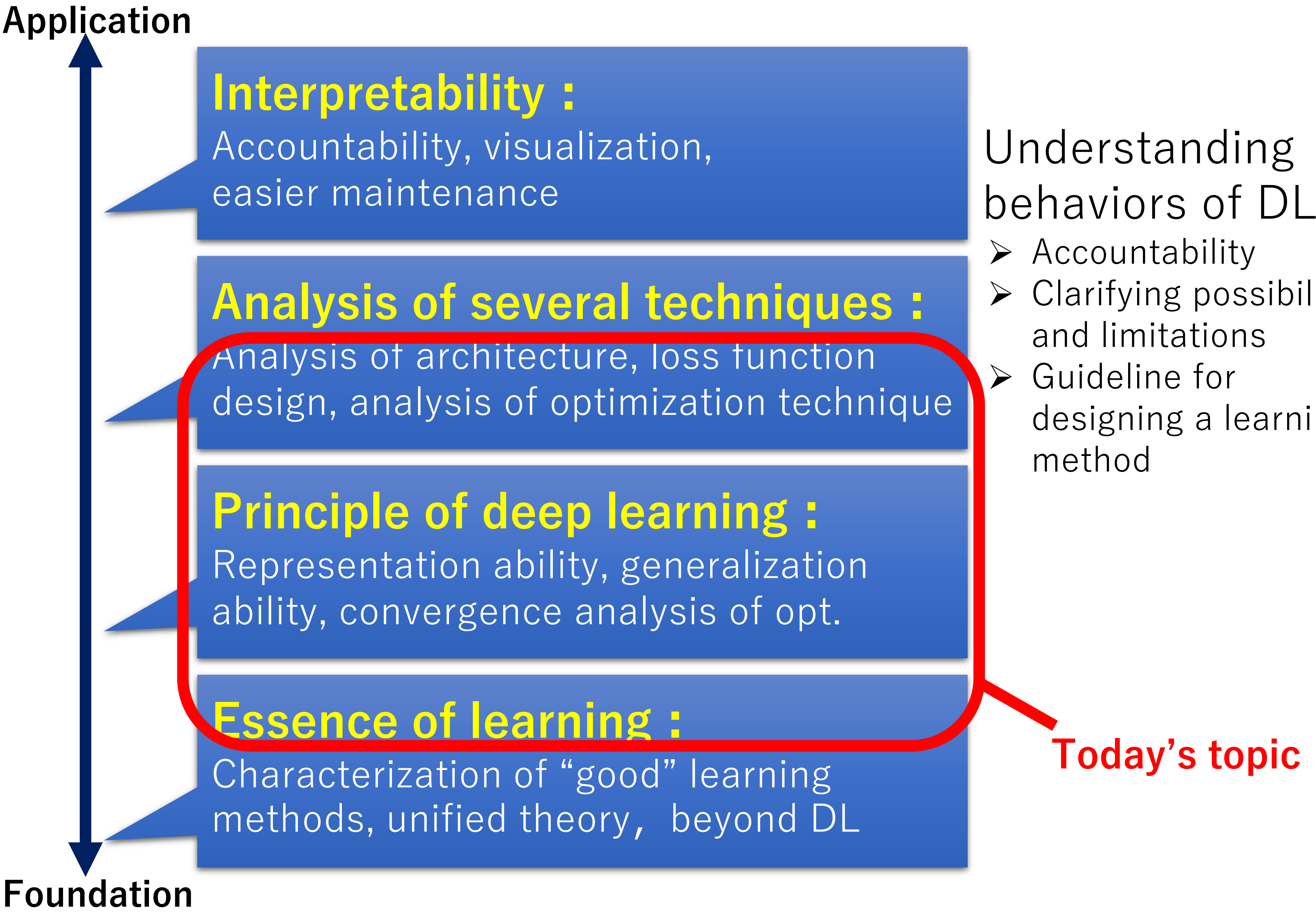
Deep learning

Several mathematicians/physicists join the ML community.

- Prob. theory
- Functional anal.
- Wasserstein geom.
- Diffusion equation
- Statistics
- Optimization
- Numerical analysis

Mathematics

Layers of deep learning theory



Application

Interpretability :
Accountability, visualization, easier maintenance

Analysis of several techniques :
Analysis of architecture, loss function design, analysis of optimization technique

Principle of deep learning :
Representation ability, generalization ability, convergence analysis of opt.

Essence of learning :
Characterization of “good” learning methods, unified theory, beyond DL

Foundation

Today’s topic

Understanding behaviors of DL

- Accountability
- Clarifying possibility and limitations
- Guideline for designing a learning method

3 issues of deep learning theory

Representation ability

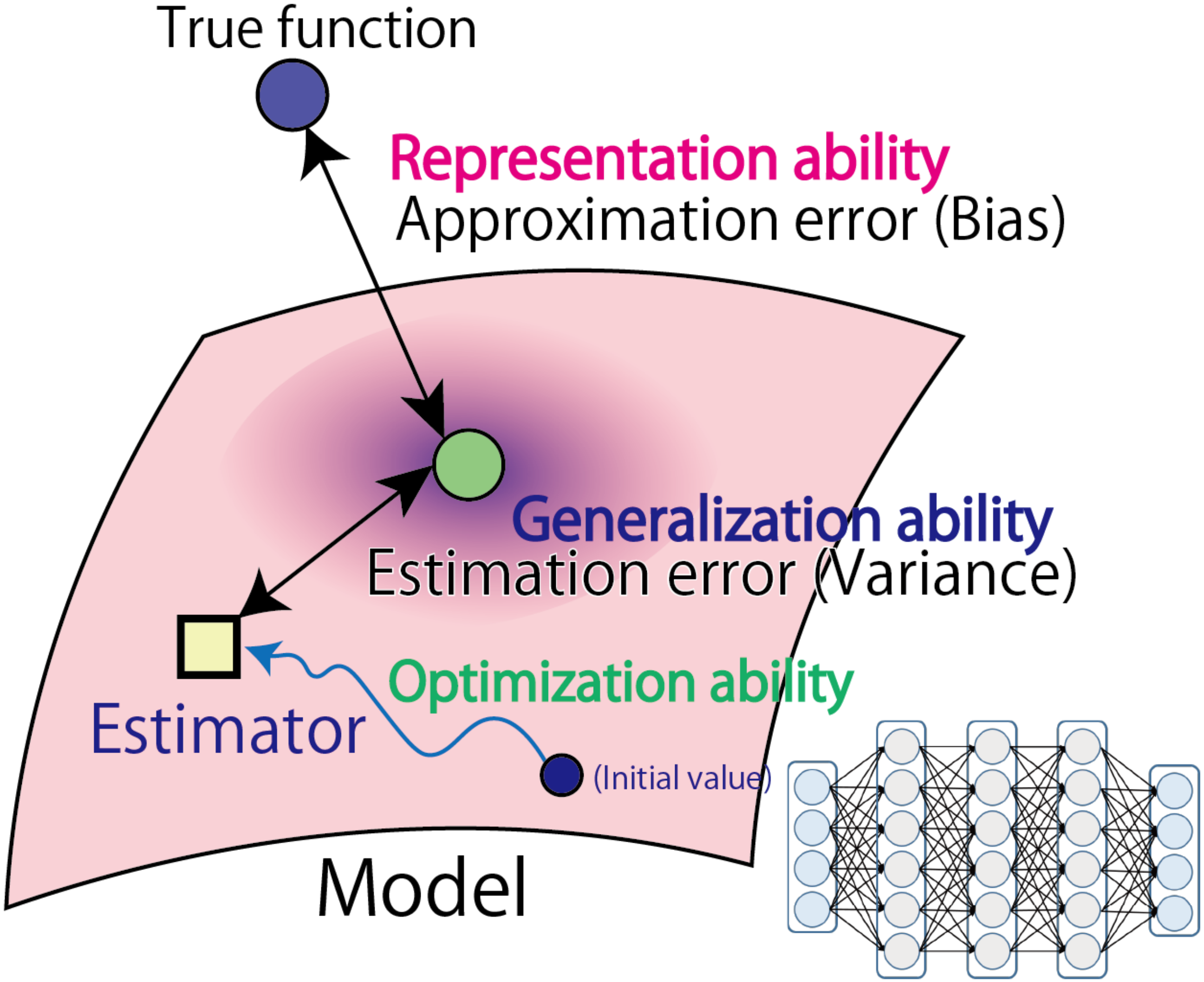
What kind of functions can DNN approximate?

Generalization ability

How well can DL generalize from finite observations?

Optimization ability

How fast can we find the optimal parameter?



True function

Representation ability

Approximation error (Bias)

Generalization ability

Estimation error (Variance)

Estimator

Optimization ability

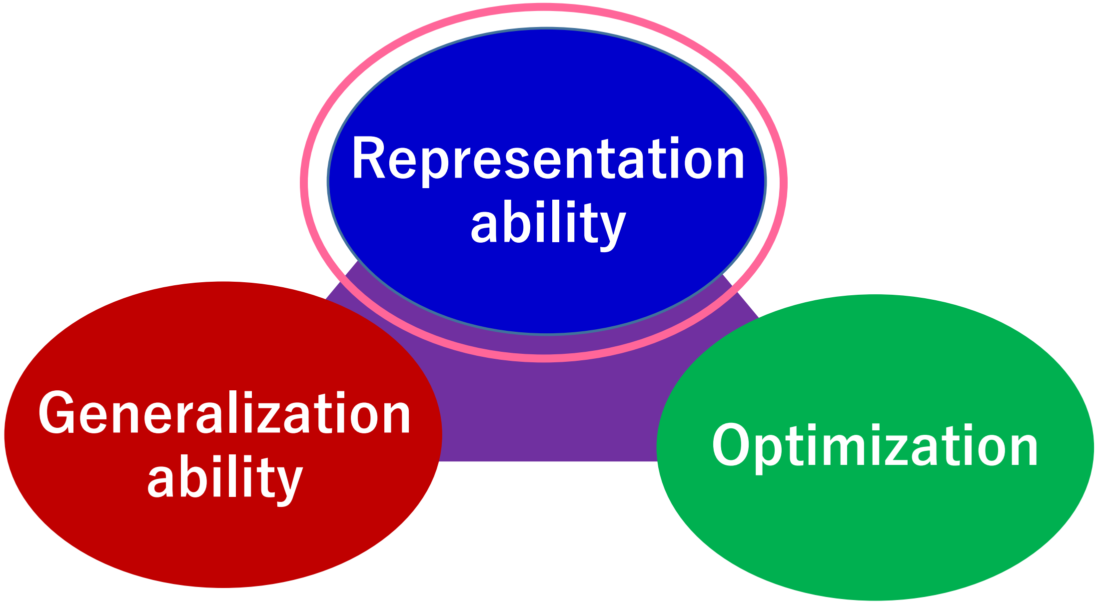
(Initial value)

Model

Outline

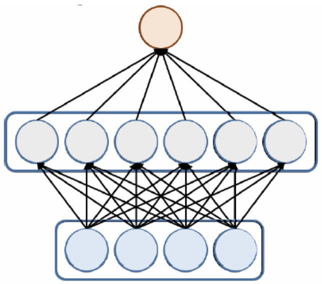
1. **Representation ability + Generalization ability**
    - ➢Universal approximator
    - ➢Depth separation
    - ➢Adaptivity of deep learning
        - ➢Inhomogeneity of smoothness
        - ➢Curse of dimensionality
    - ➢Foundation models
        - ➢Diffusion model
        - ➢Transformer
2. **Optimization ability**
    - ➢Noisy gradient descent
    - ➢Mean field Langevin
    - ➢CSQ lowerbound

Representation ability of neural networks



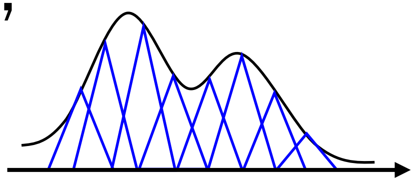
Representation ability

Universal approximator
Neural networks can approximate
“any function” with “any precision”.



2-layer NNs can approximate any function,
by increasing the number of neurons.

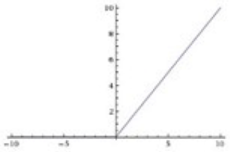
[Hecht-Nielsen,1987][Cybenko,1989]



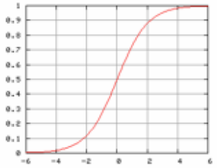
Year		Basis function	Space
1987	Hecht-Nielsen	Depending on the target	$C(R^d)$
1988	Gallant & White	Cos	$L_2(K)$
	Irie & Miyake	integrable	$L_2(R^d)$
1989	Carroll & Dickinson	Continuous sigmoidal	$L_2(K)$
	Cybenko	Continuous sigmoidal	$C(K)$
	Funahashi	Monotone & bounded	$C(K)$
1993	Mhaskar + Micchelli	Polynomial growth	$C(K)$
2015	Sonoda + Murata	**Unbounded**, admissible	$L_1(R^d), L_2(R^d)$

K is any compact set.

ReLU: $\eta(u) = \max\{u, 0\}$



Sigmoid: $\eta(u) = \frac{1}{1+\exp(-u)}$

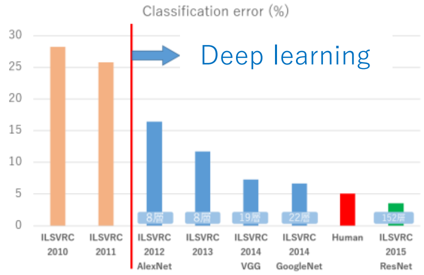


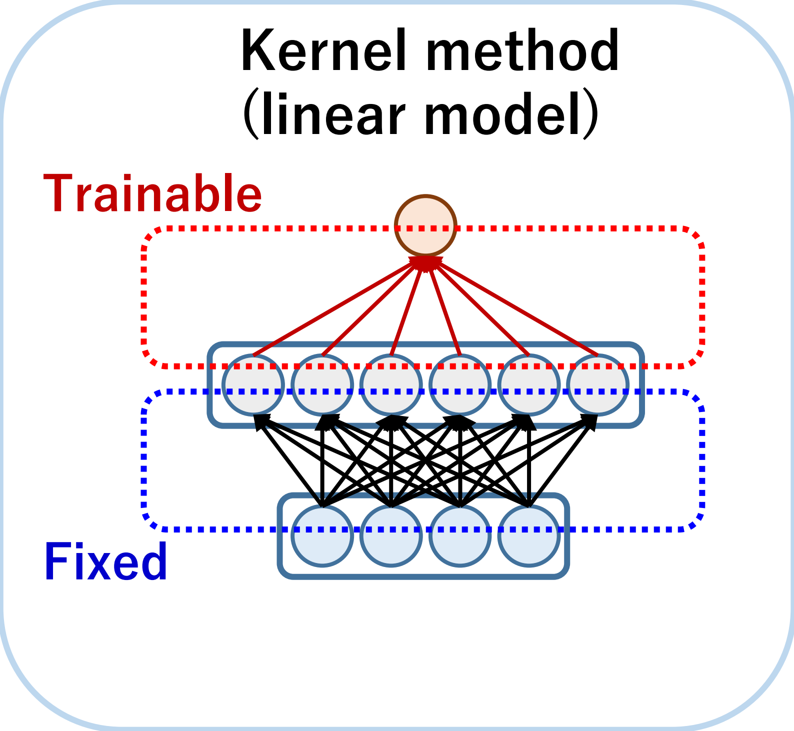
What are we missing?

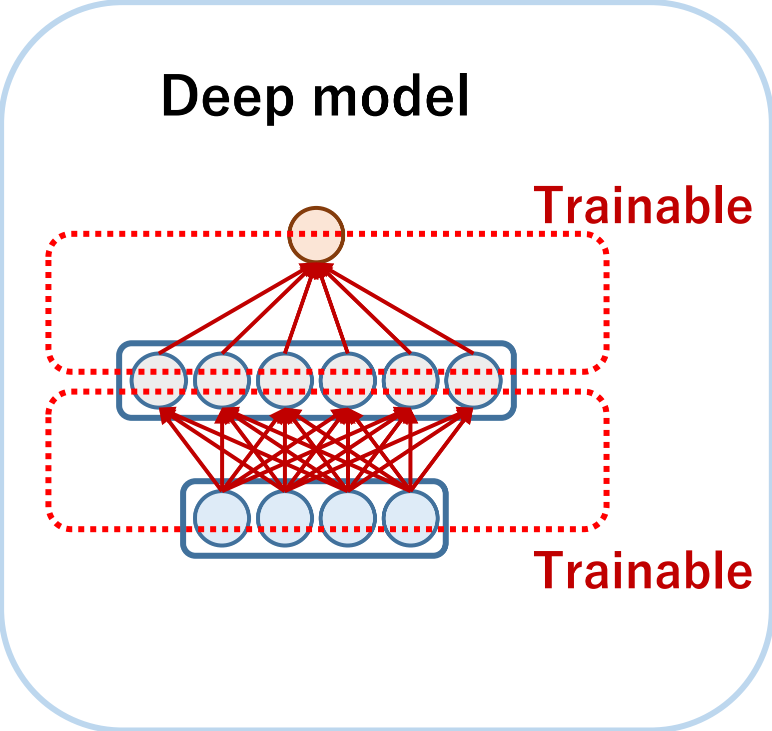
- **[Theory]** Kernel method is also a universal approximator.
- **[Practice]** DL performs better.

(in some case)

→ Why?



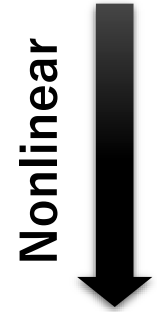

Kernel method (linear model)


Deep model

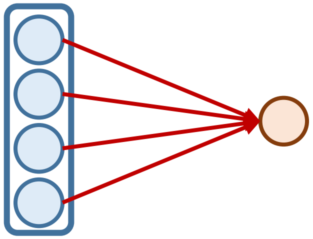
→ We compare “accuracy” of estimation/approximation.

Feature learning

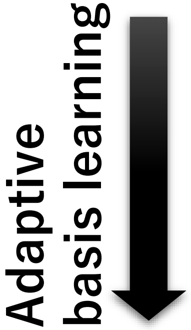
- Linear model



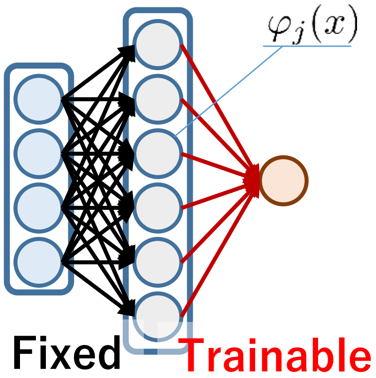
$$f(x) = \sum_{j=1}^{d} \alpha_j x_j$$



- Kernel model

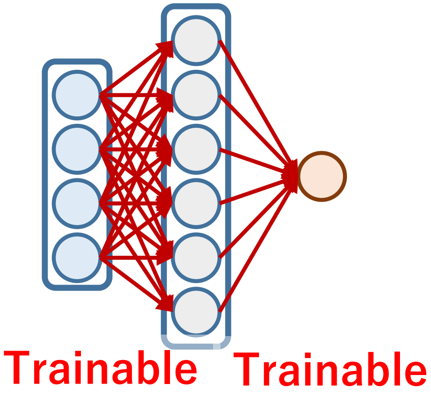


$$f(x) = \sum_{j=1}^{M} \alpha_j \underbrace{\varphi_j(x)}_{\text{Fixed}}$$

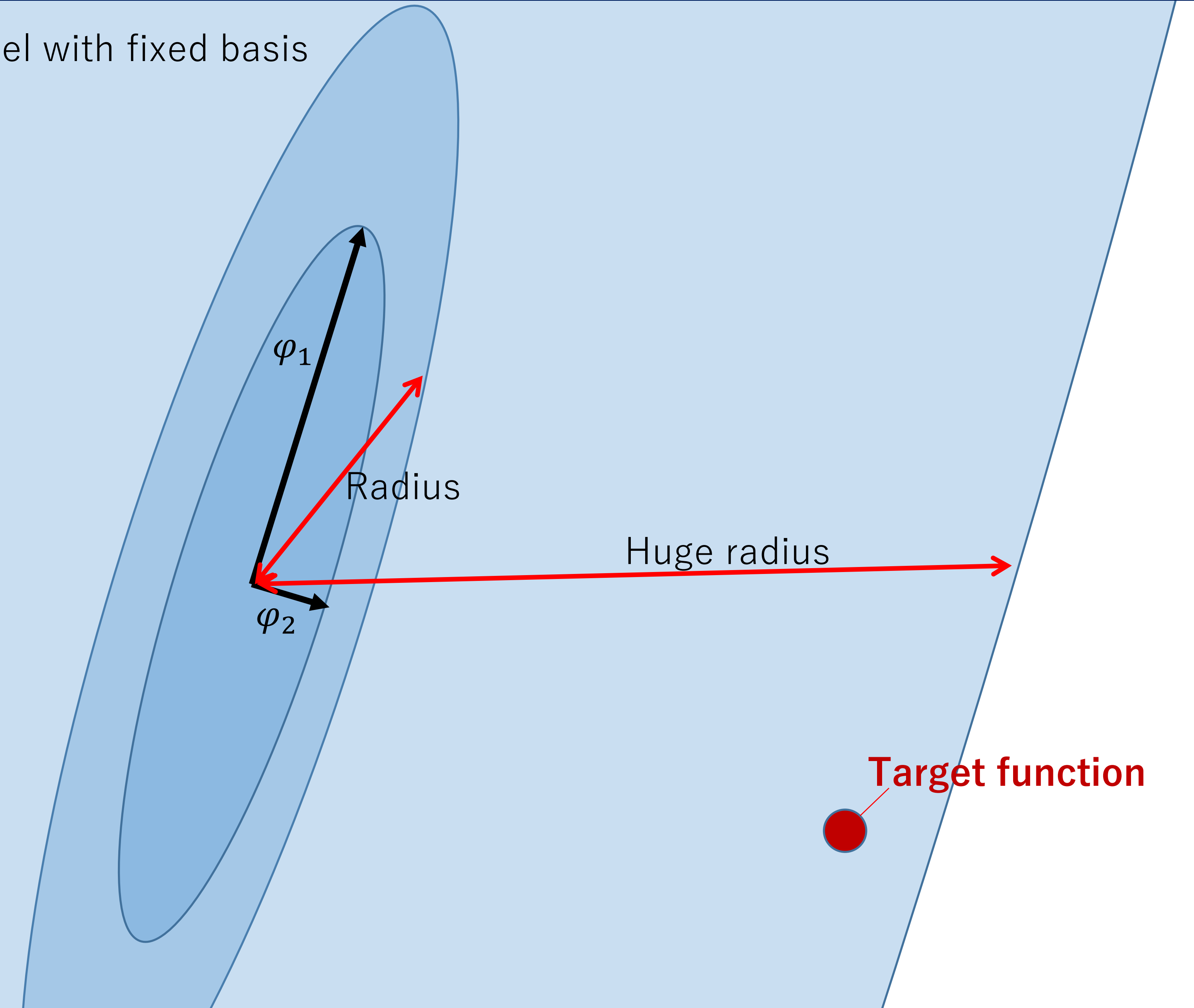


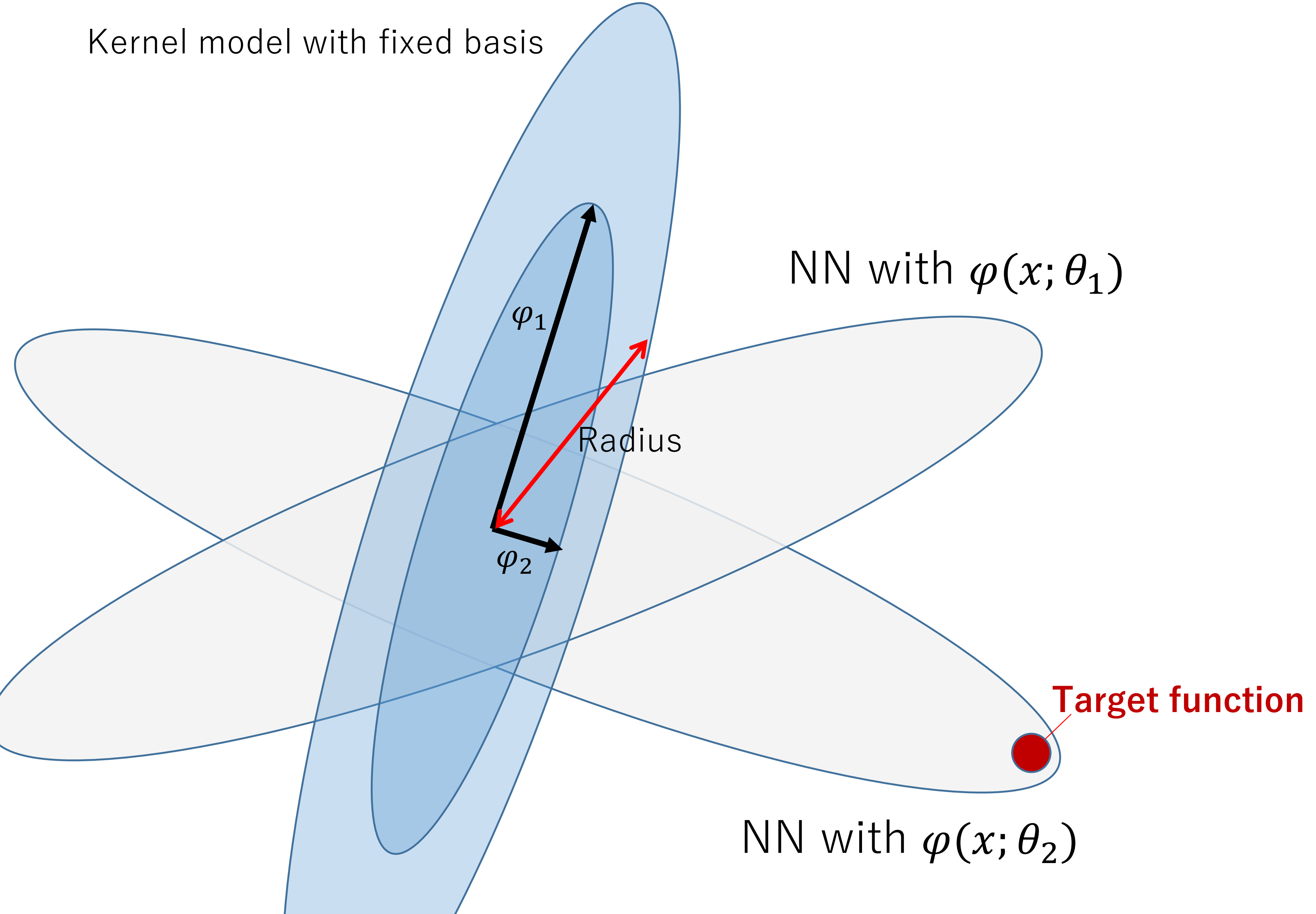
- Neural network

$$f(x) = \sum_{j=1}^{M} \alpha_j \underbrace{\varphi_j(x; \theta)}_{\text{Trainable}}$$



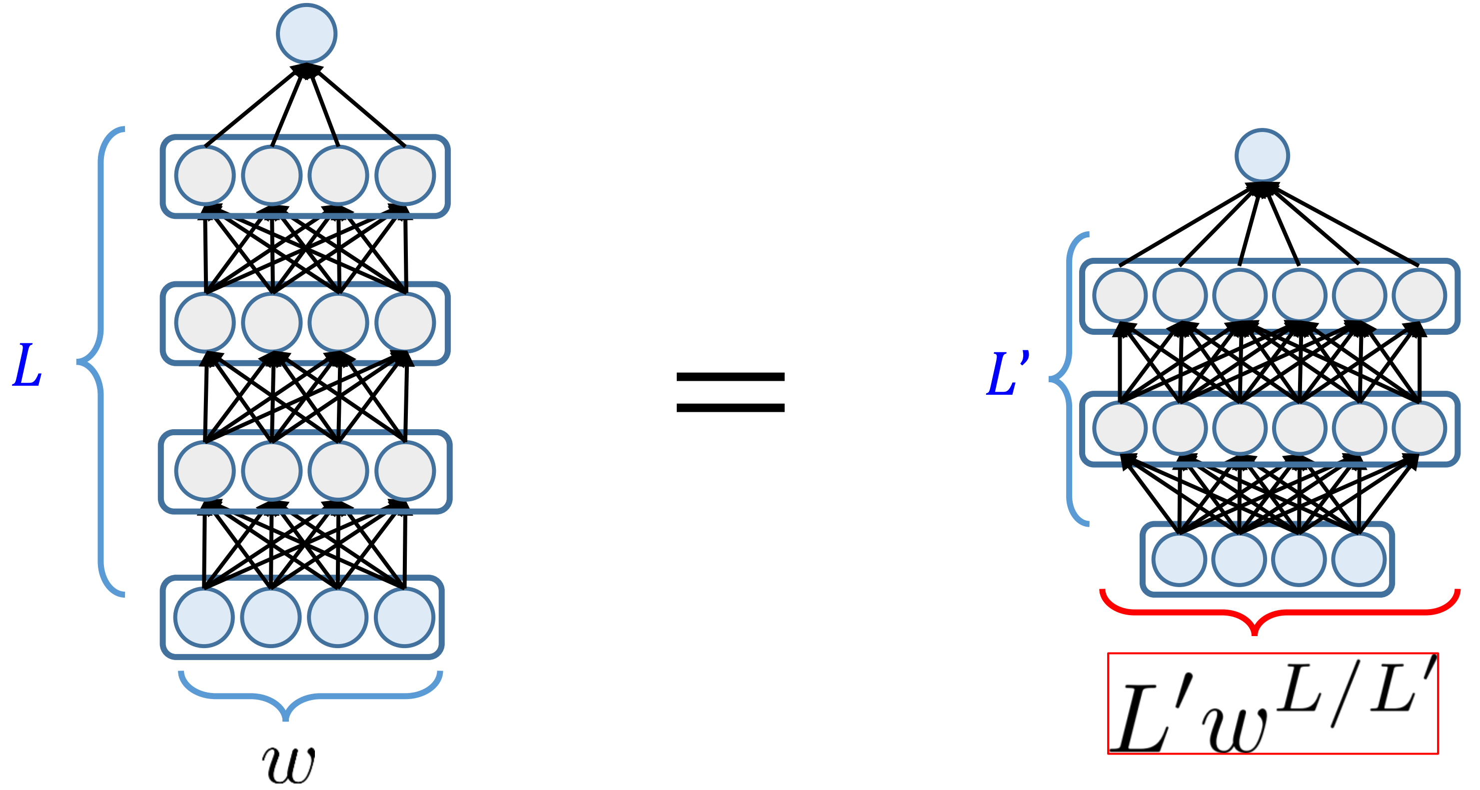
Kernel model with fixed basis




Kernel model with fixed basis
NN with $\varphi(x;\theta_1)$
φ_1
Radius
φ_2
Target function
NN with $\varphi(x;\theta_2)$

Deep network is exponentially powerful

Width vs Depth



Exponentially large width is required.

[Arora, Basu, Mianjy, Mukherjee: Understanding Deep Neural Networks with Rectified Linear Units. ICLR2018.]

Curse of dimensionality/Barron class

π: probability measure on $\mathbb{S}^{d-1} = \{x \in \mathbb{R}^d \mid \|x\| = 1\}$

- Mean field model defined by π (σ: ReLU)

$$\mathcal{H}_\pi = \left\{ \int_{\mathbb{S}^{d-1}} a(w)\sigma(w^\top x)\pi(\mathrm{d}w) \,\middle|\, \int_{\mathbb{S}^{d-1}} a(w)^2\pi(\mathrm{d}w) < \infty \right\}$$

$$\|f\|_{\mathcal{H}_\pi}^2 := \inf_a \mathbb{E}_{w\sim\pi}[|a(w)|^2]$$

where $f = \int a(w)\sigma(w^\top x)\pi(\mathrm{d}w)$

Approx. error

- **Neural network model with M neurons**

The first layer can be tuned

$$\mathcal{H}_{\mathrm{NN}}(M) = \left\{ \hat{f}(x) = \textstyle\sum_{j=1}^M r_j\sigma(u_j^\top x) \,\middle|\, r_j \in \mathbb{R},\ u_j \in \mathbb{S}^{d-1} \right\}$$

: set of NNs

$$\inf_{\hat{f}\in\mathcal{H}_{\mathrm{NN}}} \|f - \hat{f}\|_{L_2(P_X)}^2 = O(1/M) \qquad (\forall f \in \mathcal{H}_\pi)$$

- **Random feature model with M neurons**

Generated randomly
(fixed independent of the target)

$$\mathcal{H}_{\mathrm{rand}}(M) = \left\{ \hat{f}(x) = \textstyle\sum_{j=1}^M r_j\sigma(\boxed{u_j^\top} x) \,\middle|\, r_j \in \mathbb{R} \right\}$$

$$\inf_{\hat{f}\in\mathcal{H}_{\mathrm{rand}}(M)} \|f - \hat{f}\|_{L_2(P_X)}^2 \gtrsim \frac{1}{d^2 M^{2/d}} \qquad (\exists \pi, \exists f \in \mathcal{H}_\pi)$$

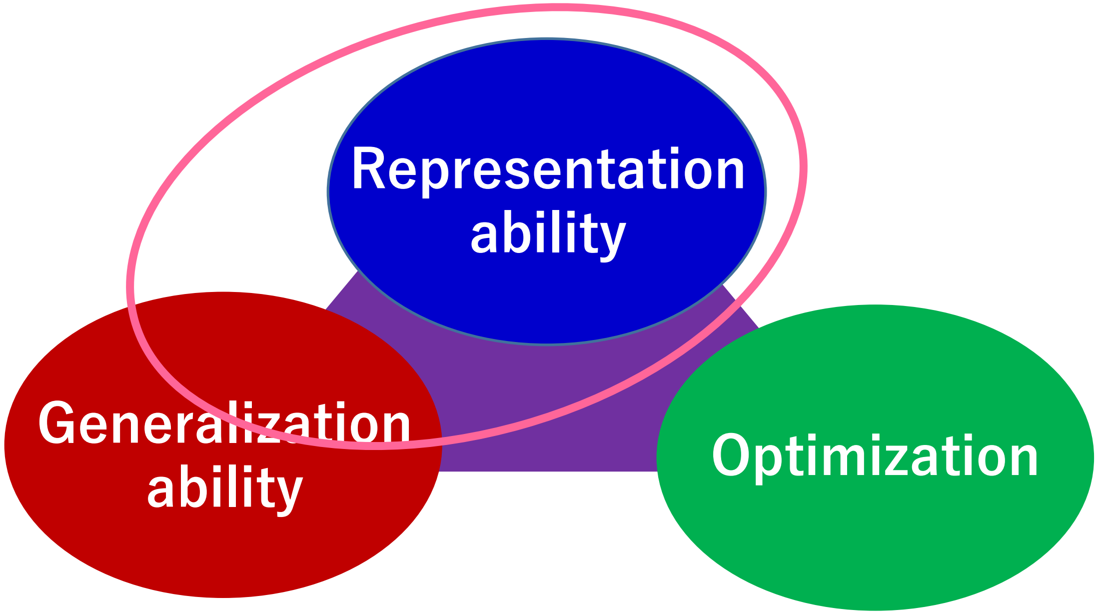
Curse of dimensionality (To obtain ϵ accuracy, $M = \epsilon^{-\Omega(d)}$ is required)

[E, Ma, Wu: A comparative analysis of optimization and generalization properties of two-layer neural network and random feature models under gradient descent dynamics. Science China Mathematics volume 63, 1235–1258 (2020)][E, Ma, Wu: A priori estimates of the population risk for two-layer neural networks. COMMUN. MATH. SCI. 17(5), 1407–1425 (2019)]

Outline

1. **Representation ability + Generalization ability**
   - Universal approximator
   - Depth separation
   - Adaptivity of deep learning
     - Inhomogeneity of smoothness
     - Curse of dimensionality
   - Foundation models
     - Diffusion model
     - Transformer
2. **Optimization ability**
   - Noisy gradient descent
   - Mean field Langevin
   - CSQ lowerbound

Analysis in nonparametric regression -Superiority of deep learning-



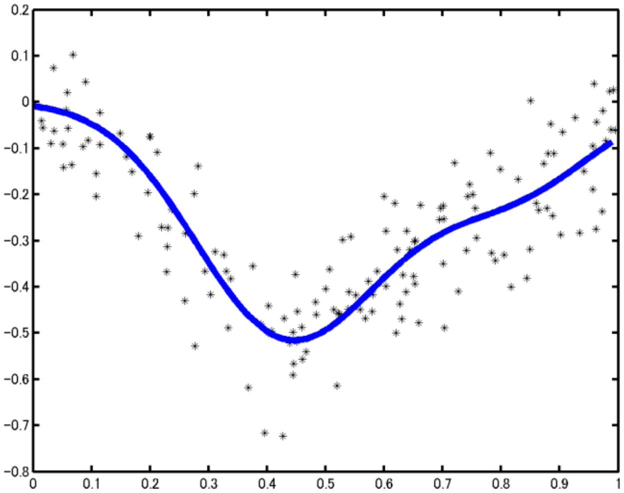
Non-parametric regression

Non-parametric regression

$$y_i = f^\circ(x_i) + \xi_i \qquad (i = 1, \dots, n)$$

where $\xi_i \sim N(0, \sigma^2)$ and $x_i \in [0,1]^d \sim P(X)$ (i.i.d.).

We estimate f° from $(x_i, y_i)_{i=1}^n$.



Estimation error:

$$\mathbb{E}[\|\hat{f} - f^\circ\|_{L_2(P)}^2] < ?$$

A similar argument can be applied to classification.

Bias-Variance decomposition

Model $\mathcal{F}$: d-dimensional parameter
n: data size

$$\hat{f} \leftarrow \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2$$

Model's degrees of freedom

$$\text{Predictive error} = \frac{\boxed{M}}{n} + [\text{Approx. error}]$$

(mean squared error)

$$\mathbb{E}[(f^\circ(X) - \hat{f}(X))^2]$$

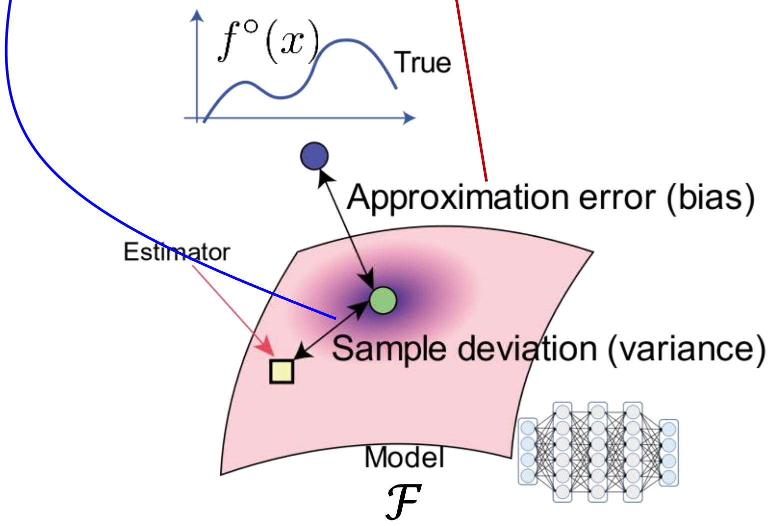
Variance

Model Bias

$$\inf_{f \in \mathcal{F}} \mathbb{E}[(f^\circ(X) - f(X))^2]$$

- **Bias-variance trade-off**
  Large model is not necessarily good
- d can be replaced by an "intrinsic" dimensionality.
- Several learning theory is reduced to evaluate the bias and variance terms.

[Gine&Nickl: Mathematical Foundations of Infinite-Dimensional Statistical Models. 2015]
[Schmidt-Hieber, 2017;Hayakawa&Suzuki, 2020]



Bias-Variance decomposition

Model $\mathcal{F}$: d-dimensional parameter

n: data size

$$\hat{f} \leftarrow \arg\min_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2$$

(covering number)

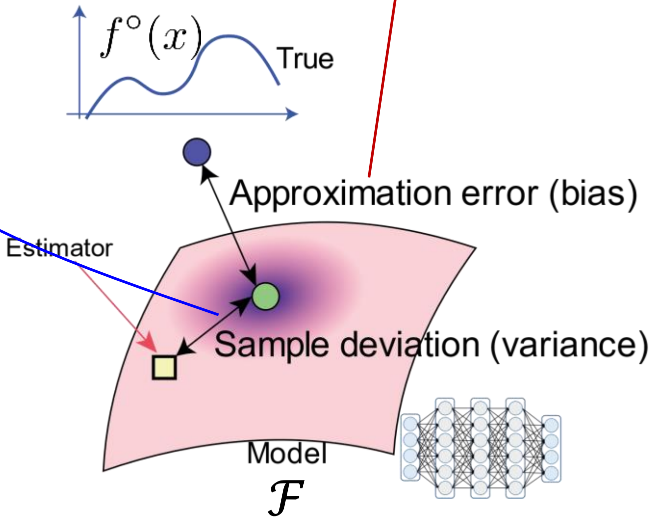
$$\mathbb{E}[(f^\circ(X) - \hat{f}(X))^2] = \frac{\log(\mathcal{N}(\mathcal{F}, \|\cdot\|_\infty, 1/n))}{n} + \inf_{f\in\mathcal{F}} \mathbb{E}[(f^\circ(X) - f(X))^2]$$

(mean squared error)

Variance

Model Bias

- **Bias-variance trade-off**
  Large model is not necessarily good
- d can be replaced by an "intrinsic" dimensionality.
- Several learning theory is reduced to evaluate the bias and variance terms.

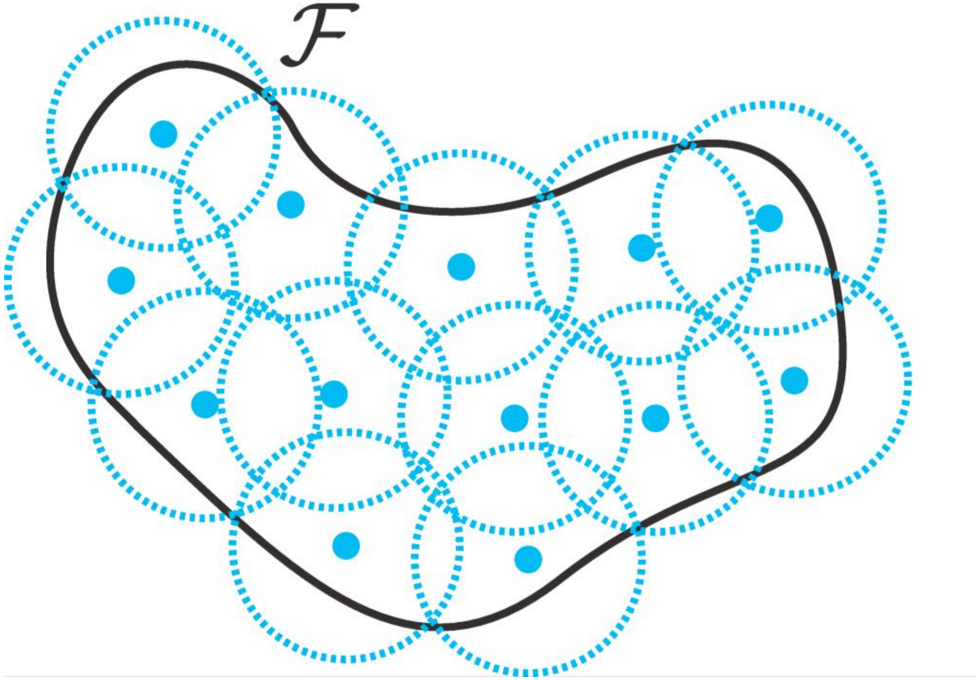


[Gine&Nickl: Mathematical Foundations of Infinite-Dimensional Statistical Models. 2015]
[Schmidt-Hieber, 2017;Hayakawa&Suzuki, 2020]

Covering number

$$\mathcal{N}(\mathcal{F}, \|\cdot\|_\infty, \epsilon)$$

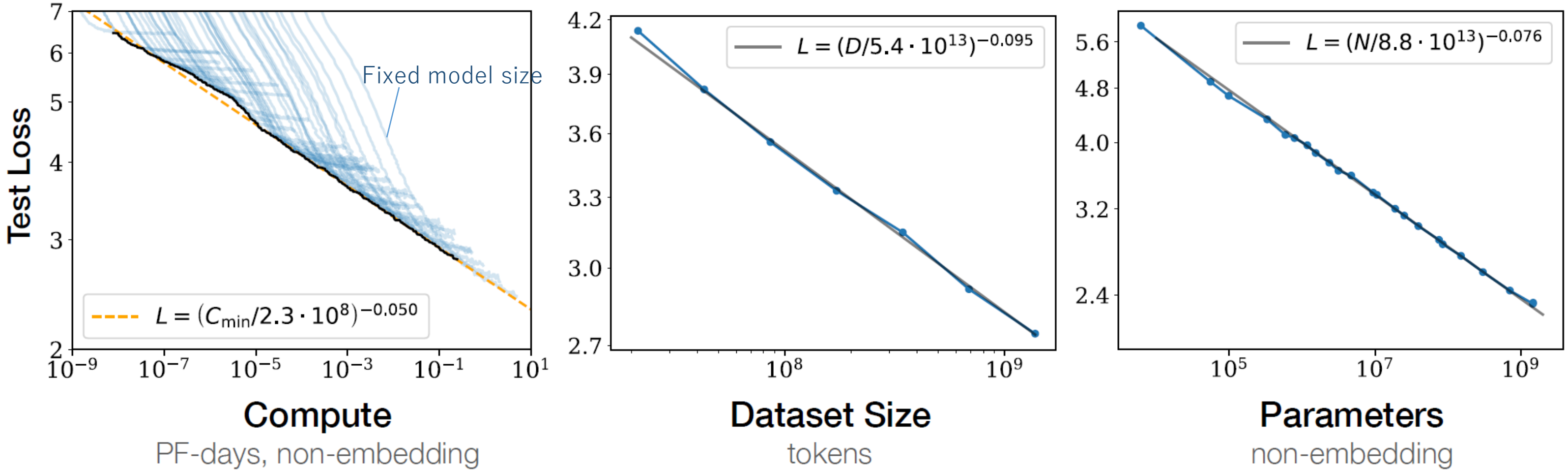
The smallest number of balls with radius ϵ measured by the norm $\|\cdot\|_\infty$ to cover the function class $\mathcal{F}$.



- The covering number measures how large the space $\mathcal{F}$ is.
- In other words, it represents “complexity” of the model.

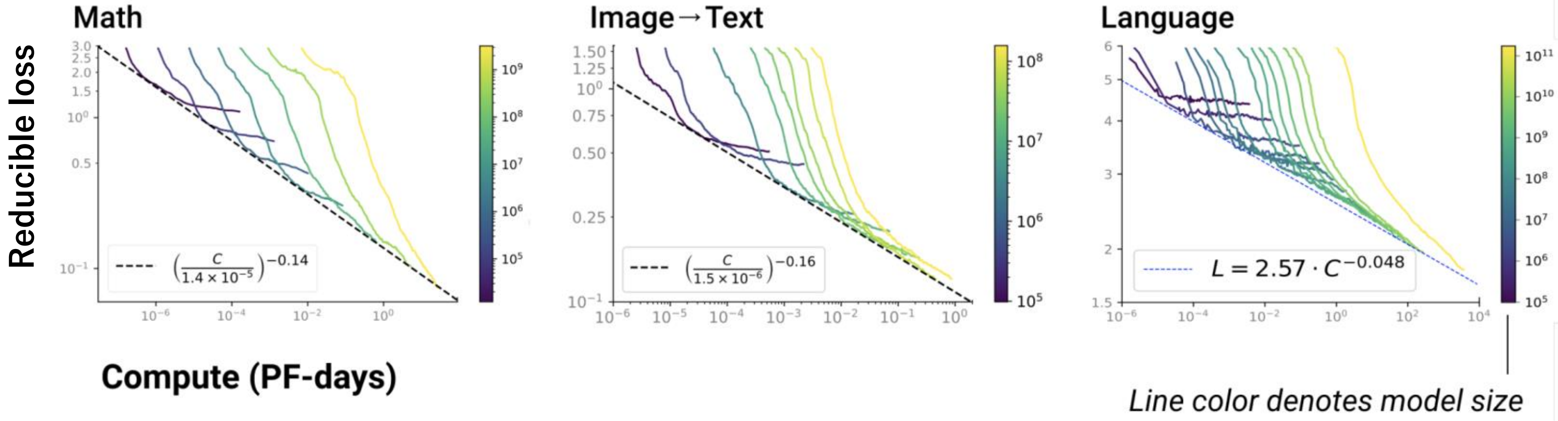
Scaling law

[Kaplan et al.: Scaling Laws for Neural Language Models, 2020]



(Data size)

[Henighan et al.: Scaling Laws for Autoregressive Generative Modeling, 2020]



$$\log(\text{Predictive error}) = -\alpha \log(n) + \log(C)$$

[Brown et al.: Language Models are Few-Shot Learners, 2020] ←Analysis of GPT-3

Analysis of kernel model

Model

(ONS in L2)

$$f^\circ(x) = \textstyle\sum_{j=1}^\infty \alpha_j \varphi_j(x)$$

Observation: $y_i = f^\circ(x_i) + \epsilon_i$

Assumption

$\mu_j \sim j^{-a}$

$\sum_{j=1}^\infty \alpha_j^2/\mu_j < \infty$

Training method

M dimensional model

$$\widehat{f}(x) = \sum_{j=1}^{M} \widehat{\alpha}_j \varphi_j(x)$$

Least squares estimator

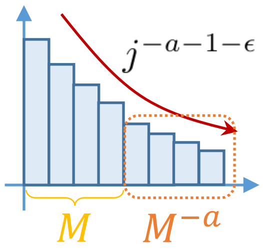
$$\min_{(\widehat{\alpha}_j)_{j=1}^M} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{M} \widehat{\alpha}_j \varphi_j(x_i) \right)^2$$

Predictive error

$$\|f^\circ - \widehat{f}\|_{L_2(P_X)}^2 \leq C \left(\overbrace{\frac{M}{n}}^{\text{Variance}} + \overbrace{M^{-a}}^{\text{Bias}} \right)$$

Optimal rate

$$n^{-\frac{a}{1+a}} \qquad (M^* = n^{\frac{1}{1+a}})$$

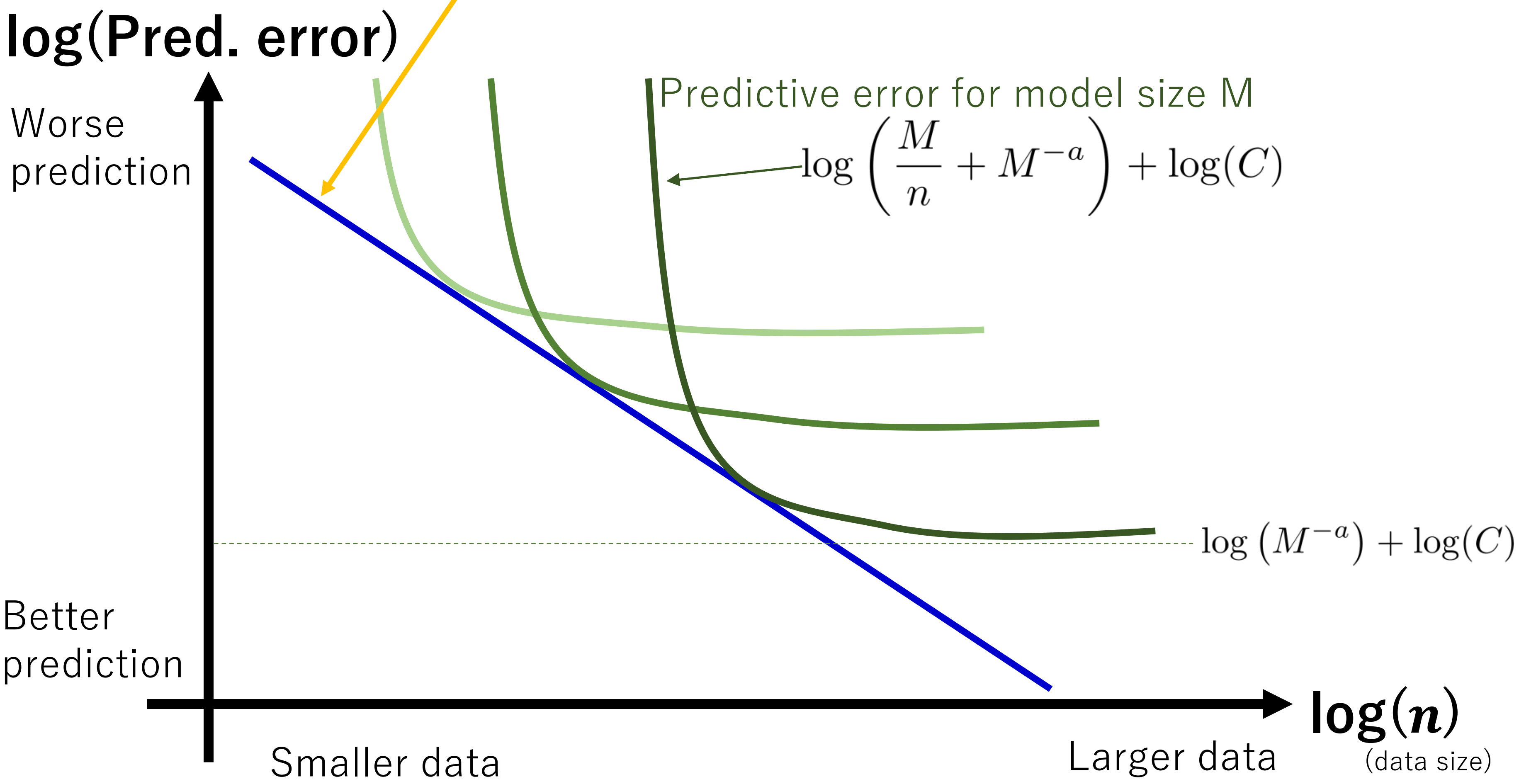


$$\log(\text{Pred. error}) = -\frac{a}{1+a}\log(n) + \log(C)$$

Variance=(dim of model)/n
Bias=L2-norm of residual

Predictive error with the optimal model size

$$\log(\text{Predictive error}) = -\frac{a}{1+a}\log(n) + \log(C)$$



Statistical learning theory of kernel methods

- Caponnetto and De Vito. Optimal Rates for the Regularized Least-Squares Algorithm. *Foundations of Computational Mathematics*, volume 7, pp.331–368 (2007).
- Steinwart and Christmann. *Support Vector Machines*. 2008.

Related recent papers

- Mei, Misiakiewicz, Montanari. Generalization error of random features and kernel methods: hypercontractivity and kernel matrix concentration. arXiv:2101.10588.
- Bordelon, Canatar, Pehlevan. Spectrum Dependent Learning Curves in Kernel Regression and Wide Neural Networks. ICML, 1024-1034, 2020.
- Canatar, Bordelon, Pehlevan. Spectral Bias and Task-Model Alignment Explain Generalization in Kernel Regression and Infinitely Wide Neural Networks. Nature Communications, volume 12, Article number: 2914 (2021).

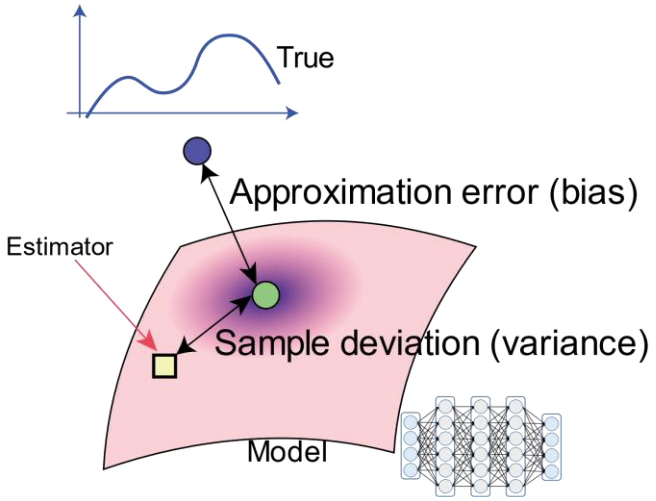
Predictive error of deep neural network

To bound the predictive error of deep learning, we evaluate

- ➢ Variance of a DL estimator (Sample variance)
- ➢ Bias of the model (Approximation error)

Typical function class

- Barron class
- Hölder class
- Sobolev class
- Besov class



Approximation theory:

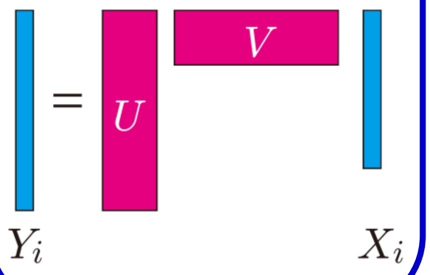
- Mhaskar: Neural networks for localized approximation of real functions. In Neural Networks for Processing III, Proceedings of the 1993 IEEE-SP Workshop, 190–196, 1993.
- Pinkus: Approximation theory of the mlp model in neural networks. Acta Numerica, 8:143–195, 1999.
- Mhaskar: Neural networks for optimal approximation of smooth and analytic functions. Neural Computation, 8(1):164–177, 1996.

Why deep?

- There are many theories…

Reduced rank regression

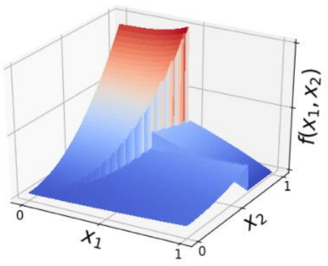
If there is low dimensional representation, deep is better.



Piece-wise smooth function

[Imaizumi&Fukumizu, 2019]

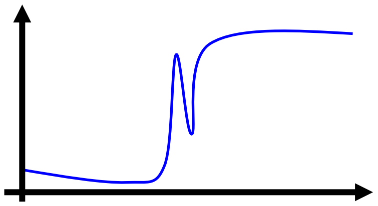
Deep is better to estimate a discontinuous function.



Besov space

[Suzuki, 2019]

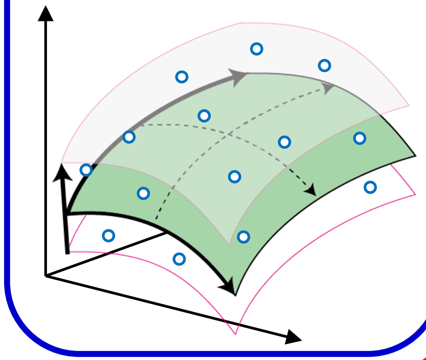
To estimate functions with non-uniform smoothness, DL is better.



Low dimensional data structure

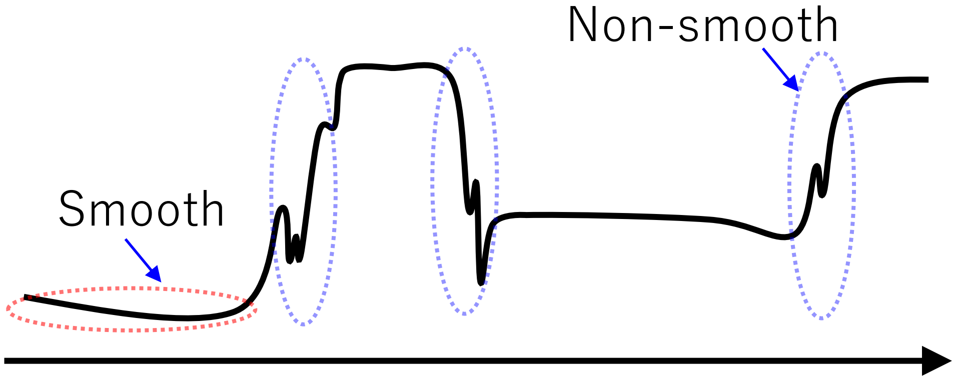
[Schmidt-Hieber, 2019] [Nakada&Imaizumi, 2019][Chen et al., 2019][Suzuki&Nitanda, 2019]

If data are distributed on low dim-manifold, DL is better.

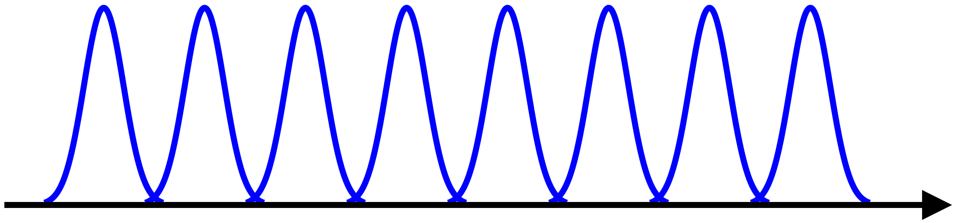


Deep	$\frac{r(M+N)}{n}$	$n^{-\frac{2s}{2s+d}} \vee n^{-\frac{\alpha}{\alpha+D-1}}$	$n^{-\frac{2s}{2s+d}}$	$n^{-\frac{2s}{2s+D}}$	Est. error
Kernel	$\frac{MN}{n}$	$\frac{1}{\sqrt{n}}$	$n^{-\frac{2s-2d(1/p-1/2)_+}{2s+d-2d(1/p-1/2)_+}}$	$n^{-\frac{2(s-D/p+d/2)}{2(s-D/p+d/2)+d}}$ $\vee\, n^{-\frac{2s}{2s+D}}$	

Typical situation

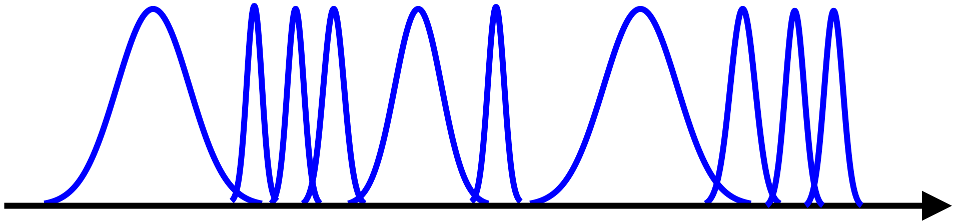


- Approximation by Gaussian kernel

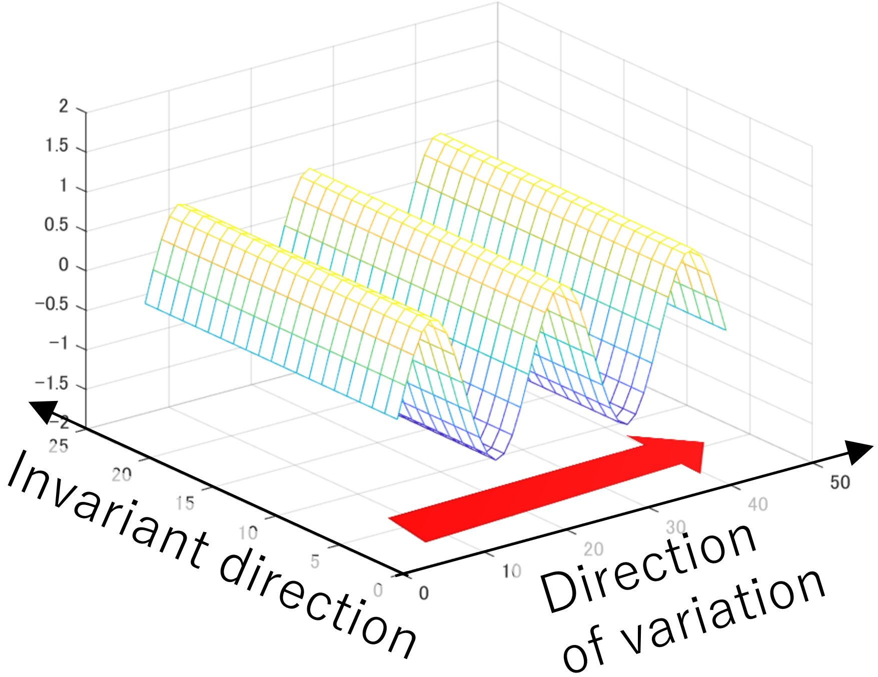


Same Gaussian width→Less efficient

- Approximation by NN



Different resolution depending on location
→More efficient



- Approximation by Gaussian kernel

  Kernel method cannot specify informative directions.
  →Curse of dimensionality

- Approximation by NN

  Informative directions is extracted in internal layers.
  →Avoids curse of dimensionality

Hölder, Sobolev, Besov space

$\Omega = [0,1]^d \subset \mathbb{R}^d$

- **Hölder space** $(\mathcal{C}^\beta(\Omega))$

$$\|f\|_{\mathcal{C}^\beta} = \max_{|\alpha| \leq m} \|\partial^\alpha f\|_\infty + \max_{|\alpha| = m} \sup_{x \in \Omega} \frac{|\partial^\alpha f(x) - \partial^\alpha f(y)|}{|x-y|^{\beta - m}}$$

- **Sobolev space** $(W_p^k(\Omega))$

$$\|f\|_{W_p^k} = \left(\sum_{|\alpha| \leq k} \|D^\alpha f\|_{L^p(\Omega)}^p \right)^{\frac{1}{p}}$$

- **Besov space** $(B_{p,q}^s(\Omega))$ $(0 < p, q \leq \infty, 0 < s \leq m)$

Spatial inhomogeneity

$$\omega_m(f,t)_p := \sup_{\|h\| \leq t} \left\| \sum_{j=0}^m (-1)^{m-j} \binom{m}{j} f(\cdot + jh) \right\|_{L^p(\Omega)},$$

$$\|f\|_{B_{p,q}^s(\Omega)} = \|f\|_{L^p(\Omega)} + \left(\int_0^\infty [t^{-s} \omega_m(f,t)_p]^q \frac{\mathrm{d}t}{t} \right)^{1/q}.$$

Smoothness

Hölder, Sobolev, Besov space

$\Omega = [0,1]^d \subset \mathbb{R}^d$

- **Hölder space** $(\mathcal{C}^{\beta}(\Omega))$

$$\|f\|_{\mathcal{C}^{\beta}} = \max_{|\alpha| \leq m} \|\partial^{\alpha} f\|_{\infty} + \max_{|\alpha| = m} \sup_{x \in \Omega} \frac{|\partial^{\alpha} f(x) - \partial^{\alpha} f(y)|}{|x - y|^{\beta}}$$

Intuition

Smoothness

$$\|f\|_{B^{s}_{p,q}(\Omega)} = \|f\|_{L^{p}(\Omega)} + \|D^{s} f\|_{L^{p}(\Omega)}$$

Uniformity of smoothness

- **Besov space** $(B^s_{p,q}(\Omega))$ $(0 < p, q \leq \infty, 0 < s \leq m)$

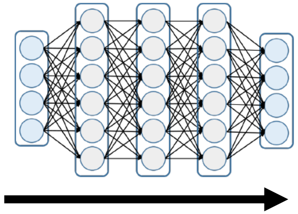
Spatial inhomogeneity

$$\omega_m(f,t)_p := \sup_{\|h\| \leq t} \left\| \sum_{j=0}^{m} (-1)^{m-j} \binom{m}{j} f(\cdot + jh) \right\|_{L^p(\Omega)},$$

$$\|f\|_{B^s_{p,q}(\Omega)} = \|f\|_{L^p(\Omega)} + \left(\int_0^{\infty} [t^{-s} \omega_m(f,t)_p]^q \frac{\mathrm{d}t}{t} \right)^{1/q}.$$

Smoothness

Deep learning has adaptivity



- DL constructs basis function "adaptively".
  →Efficient learning
- Shallow learning should use "redundant" model.
  →Inefficient learning (affected by redundant noise)

$f^\circ \in B^s_{p,q}([0,1]^d)$: "Besov space"

[Suzuki: Adaptivity of deep ReLU network for learning in Besov and mixed smooth Besov spaces: optimal rate and curse of dimensionality, ICLR2019]

Linear estimator (Shallow method)

Deep learning

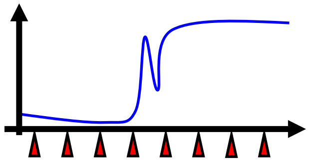
e.g., kernel ridge regression: $\hat{f}(x) = K_{x,X}(K_{X,X} + \lambda \mathrm{I})^{-1} Y$

$$n^{-\frac{2s-2d(1/p-1/2)_+}{2s+d-2d(1/p-1/2)_+}} \gg n^{-\frac{2s}{2s+d}}$$

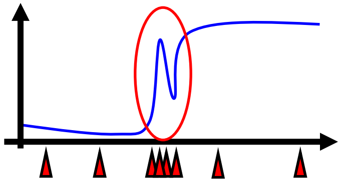
Suboptimal

Optimal

(n: sample size, p: uniformity of smoothness, s: smoothness)



Uniform resolution



Adaptive resolution

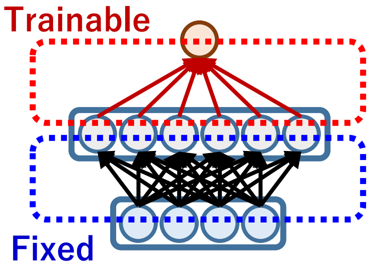
Convergence rate of estimation error (mean squared error $\mathrm{E}\left[\|\hat{f} - f^*\|^2\right]$)

- The rate of error decrease as the sample size $n \to \infty$.

Linear estimator

"Shallow" learning methods

Kernel ridge regression:



$$\hat{\beta} = \underset{\beta \in \mathbb{R}^\infty}{\arg\min} \sum_{i=1}^{n} (y_i - \beta^\top \psi(x_i))^2 + \lambda \beta^\top \beta$$

$\psi : \mathcal{X} \to \mathbb{R}^\infty$ (feature map)

Fixed

$$K_{X,X} = (\psi(x_i)^\top \psi(x_j))_{i,j=1}^{n,n}$$

Gram matrix (kernel function)

$$\hat{f}(x) = K_{x,X}(K_{X,X} + \lambda \mathrm{I})^{-1}\underline{Y}$$

(see also [Imaizumi&Fukumizu, 2019])

Nadaraya-Watson estimator:

$$\hat{f}(x) = \frac{\sum_{i=1}^{n} k(x_i, x)\underline{y_i}}{\sum_{i=1}^{n} k(x_i, x)}$$

Linear estimator: linear to the observation $Y = (y_i)_{i=1}^{n}$.

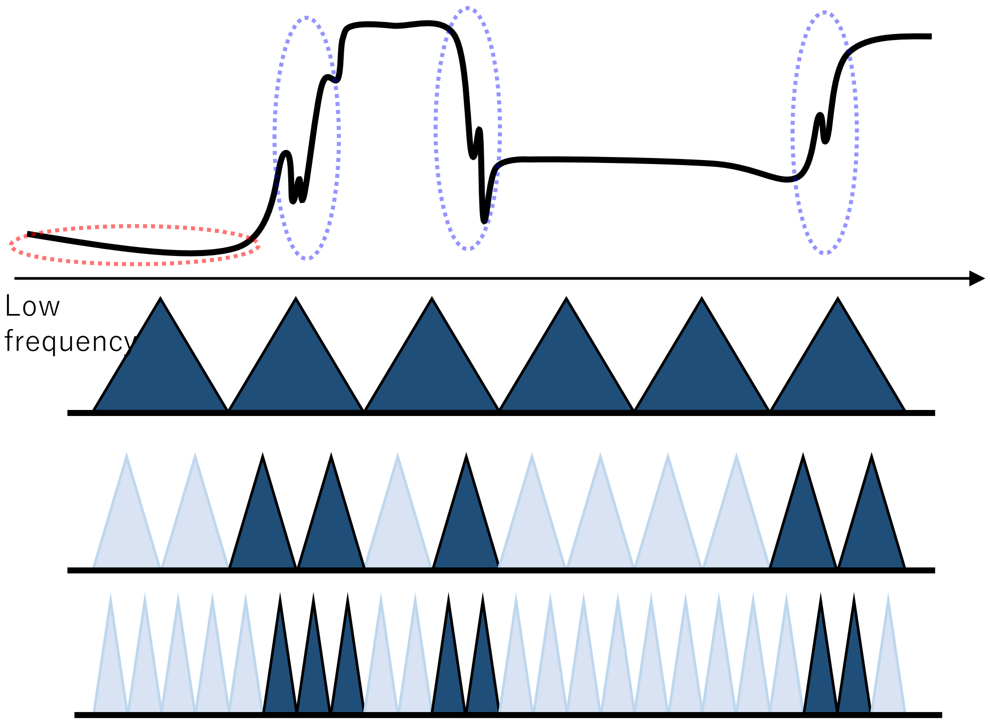
$$X_n = (x_1, \ldots, x_n)$$

$$\hat{f}(x) = \sum_{i=1}^{n} \varphi_i(x; X_n)\underbrace{y_i}_{\text{linear}}$$

Example

- Kernel ridge estimator
- Sieve estimator
- Nadaraya-Watson estimator
- k-NN estimator

Relation to sparse estimation

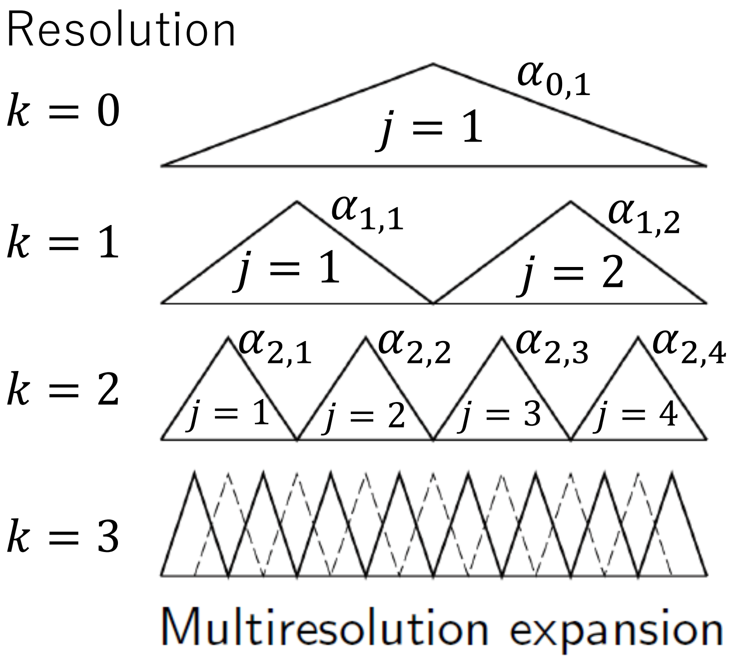


Sparse linear combination of wavelet basis functions is effective to capture input-dependent smoothness

$$f = \sum_{k\in\mathbb{N}_+} \alpha_k \phi_k$$

Wavelet basis decomposition

Wavelet basis



Multiresolution expansion

$$\|f\|_{B^s_{p,q}} = \left(\sum_{k\in\mathbb{N}_+} |\alpha_k|^p\right)^{1/p} \quad (0 < p)$$

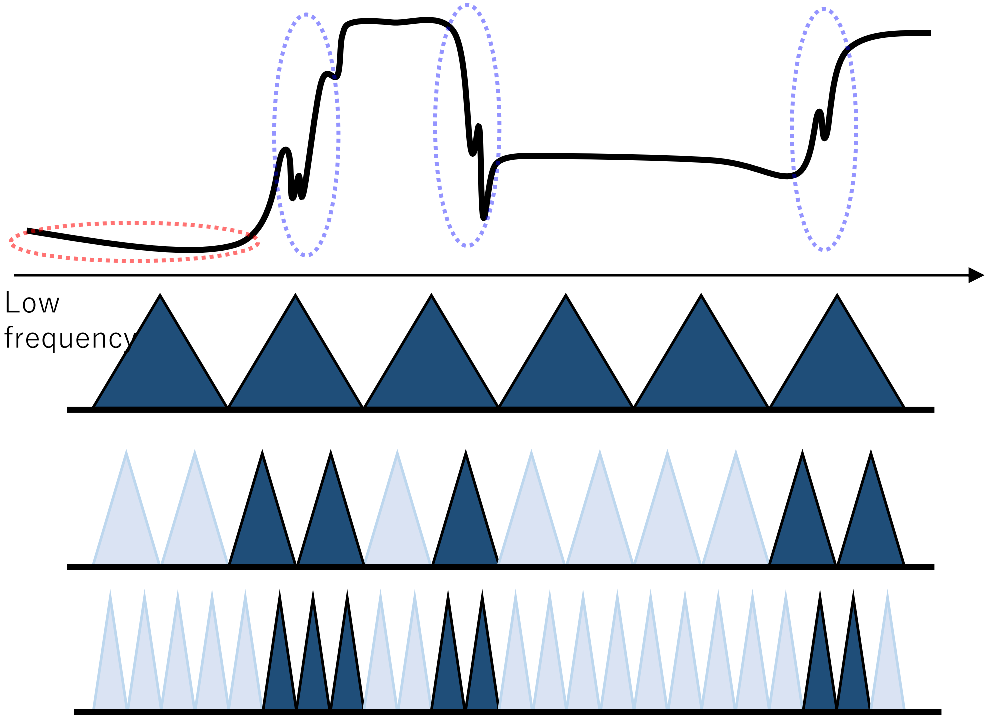
(informal)

$$\|f\|_{B^s_{p,q}} \simeq \left[\sum_{k=0}^{\infty}\{2^{sk}(2^{-kd}\sum_{j\in J(k)}|\alpha_{k,j}|^p)^{1/p}\}^q\right]^{1/q}$$

Small p = Sparse coefficient

Non-uniform smoothness over the space

Relation to sparse estimation



Sparse linear combination of wavelet basis functions is effective to capture input-dependent smoothness

$$f = \sum_{k \in \mathbb{N}_+} \alpha_k \phi_k$$

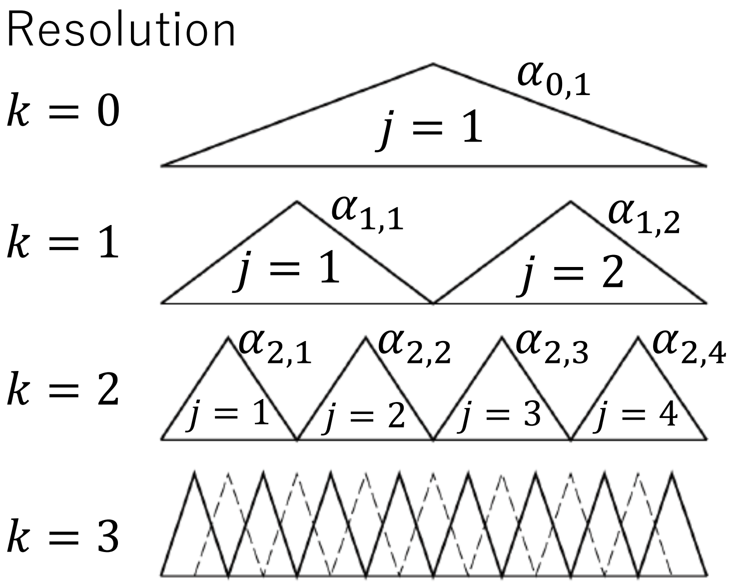
Wavelet basis decomposition

Small p = Sparse coefficient

→

Non-uniform smoothness over the space

Wavelet basis



solution expansion

$\|f$ … $|\alpha_k|^p\Big)^{1/p}$ $\quad (0 < p)$

$\mathbb{N}_+$

$$\|f\|_{B^s_{p,q}} \simeq \left[\sum_{k=0}^{\infty} \{2^{sk}(2^{-kd}\sum_{j \in J(k)} |\alpha_{k,j}|^p)^{1/p}\}^q\right]^{1/q}$$

Wavelet-shrinkage

Proof strategy

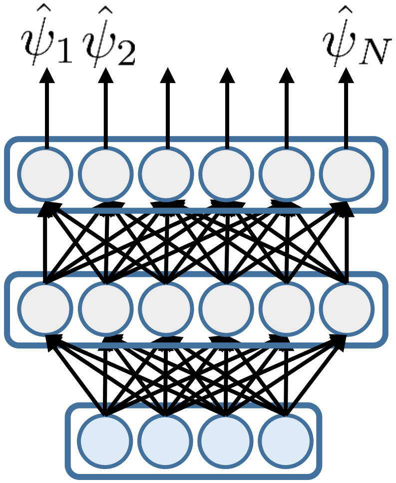
- **Step 1** : Find basis function expansion.

$$f^\circ \in \mathcal{F} \quad \Rightarrow \quad f^\circ(x) = \sum_{i=1}^{\infty} \alpha_i \psi_i(x)$$

$$f^\circ = \sum_{i=1}^{N} \alpha_i \psi_i + \underbrace{\cancel{\sum_{i=N+1}^{\infty} \alpha_i \psi_i}}_{\boxed{\|\cdot\|_{L^2} \leq N^{-s/d}}}$$

∵ Adaptive approximation by B-Spline [DeVore & Popov, 1988; Dung, 2011]

- **Step 2** : Approximate basis functions.



$\psi_i \simeq \hat{\psi}_i$: Approximation by DNN.

$$\Rightarrow \quad \check{f} = \sum_{i=1}^{N} \alpha_i \hat{\psi}_i$$: linear combination.

- **Step 3:** Combine the bounds of Step 1 and 2.

$$\|f^\circ - \check{f}\|_{L^2} \leq \sum_{i=1}^{N} |\alpha_i| \underbrace{\|\psi_i - \hat{\psi}_i\|_{L^2}}_{\leq O(e^{-L})} + \underbrace{\|\sum_{i=N+1}^{\infty} \alpha_i \psi_i\|_{L^2}}_{\leq N^{-s/d}}$$

$$\lesssim N^{-2s/d}$$

Bias variance decomposition

[Local Rademacher complexity bound]

$$\mathrm{E}[\|f^{\mathrm{o}} - \hat{f}\|_{L^2(P_X)}^2] \lesssim \underbrace{\frac{S[L\log(BW) + \log(Ln)]}{n}}_{\text{Variance}} + \underbrace{\inf_{f\in\mathcal{F}(L,W,S,B)} \|f - f^{\mathrm{o}}\|_{L^2(P_X)}^2}_{\text{Bias}}$$

Classic analysis on nonparametric regression.
See [Schmidt-Hieber, 2019; Hayakawa&Suzuki,2020] for the analysis of DNN.

Depth

Width

Sparseness
(# of non-zero parameters)

Upper bound of absolute value
of each parameter

$$L = O(\log(N)), W = O(N), S = O(N\log(N)), B = O(N^{(d/p-s)_+})$$

$$\boxed{\text{Variance} = \frac{N\log(N)^3}{n}}$$

$$\boxed{\text{Bias} = N^{-2s/d}}$$

Choose the network size N to balance bias and variance:
$N = \tilde{O}(n^{2d/(2s+d)})$.

$$\boxed{\text{Predictive error} = n^{-\frac{2s}{2s+d}}\log(n)^3}$$

Hardness: Convex hull argument

A function with a property that is destroyed by convex combination is hard to estimate by linear estimators.

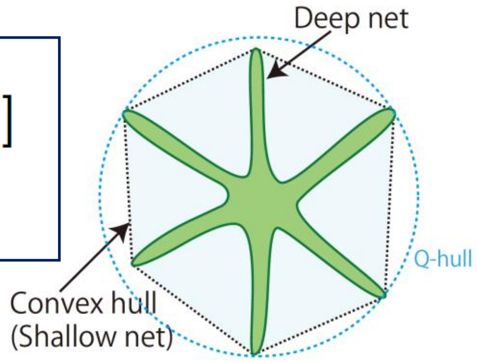
e.g., “Spatial inhomogeneity of smoothness”

Theorem

$$\inf_{\hat f:\text{Linear}} \sup_{\underline{\underline{f^{\mathrm{o}}\in\mathcal{F}}}} \mathrm{E}[\|\hat f - f^{\mathrm{o}}\|_{L_2(P)}^2] = \inf_{\hat f:\text{Linear}} \sup_{\underline{\underline{f^{\mathrm{o}}\in\mathrm{conv}(\mathcal{F})}}} \mathrm{E}[\|\hat f - f^{\mathrm{o}}\|_{L_2(P)}^2]$$

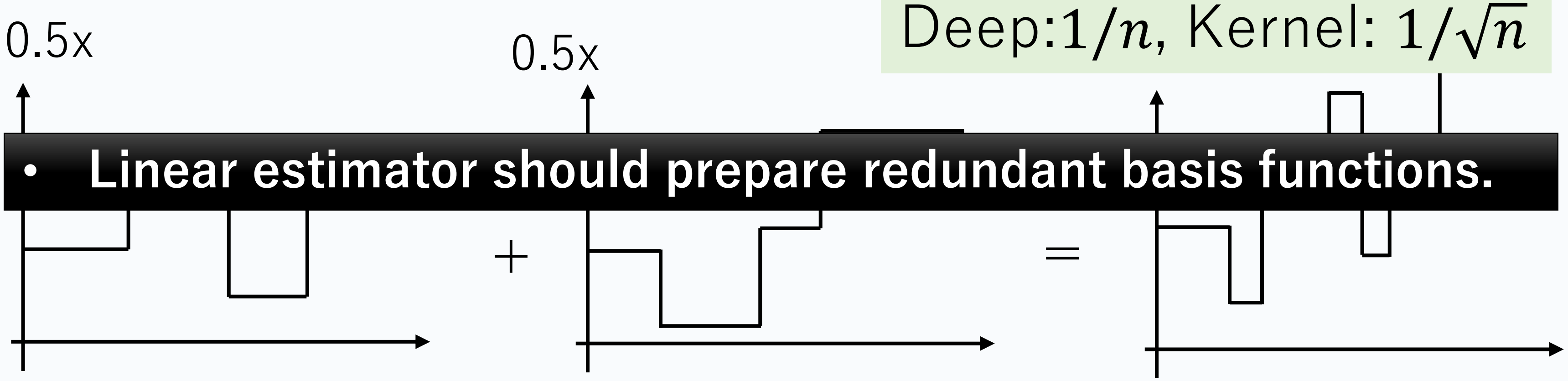
(This can be extended to Q-hull for the fixed design setting)

[Donoho & Johnstone, 1994] [Satoshi Hayakawa and Taiji Suzuki: 2020]



Ex. Piecewise constant function with 3 jumps.

Deep:$1/n$, Kernel: $1/\sqrt{n}$

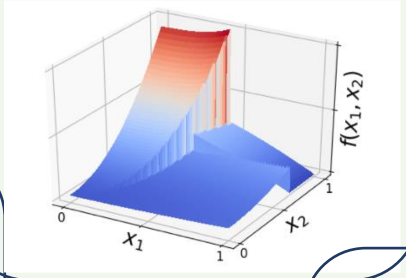


- **Linear estimator should prepare redundant basis functions.**

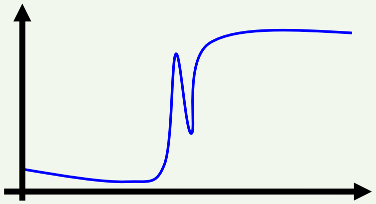
3 jumps **3** jumps **6** jumps

General theory

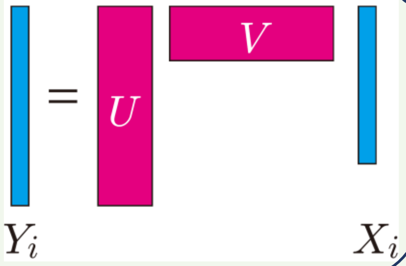
Piece-wise smooth



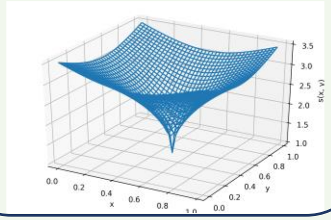
Besov space



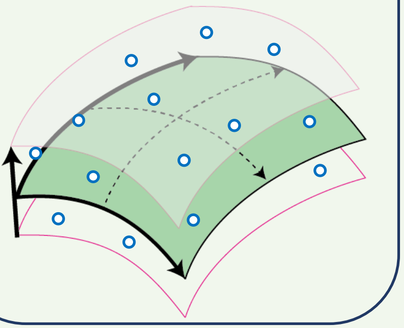
Reduced rank regression



Variable smooth Besov space



Low dim. data



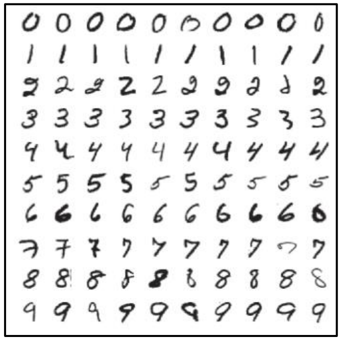
Non-convexity sparsity

Curse of dimensionality

Estimation error bound :

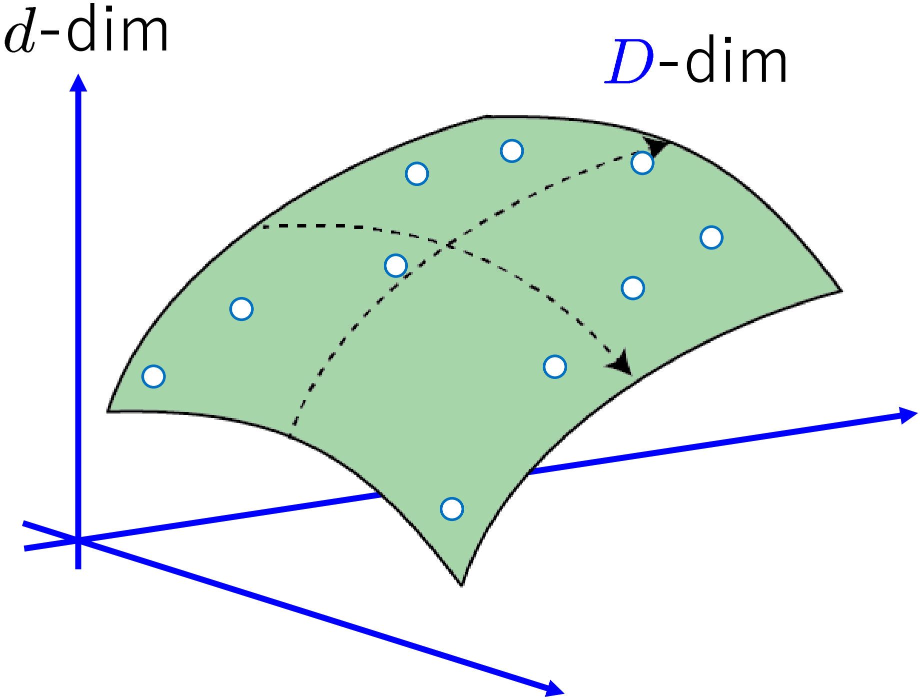
$$n^{-\frac{2s}{2s+d}}$$

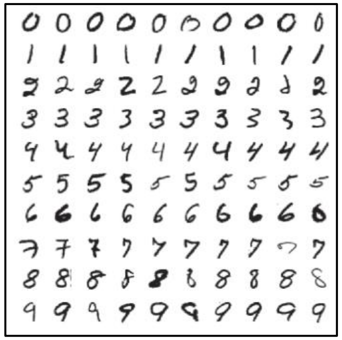
→ **Curse of dimensionality**



MNIST: 784 dim/13.4 intrinsic-dim
[Facco et al. 2017]

Dimensionality: Manifold regression





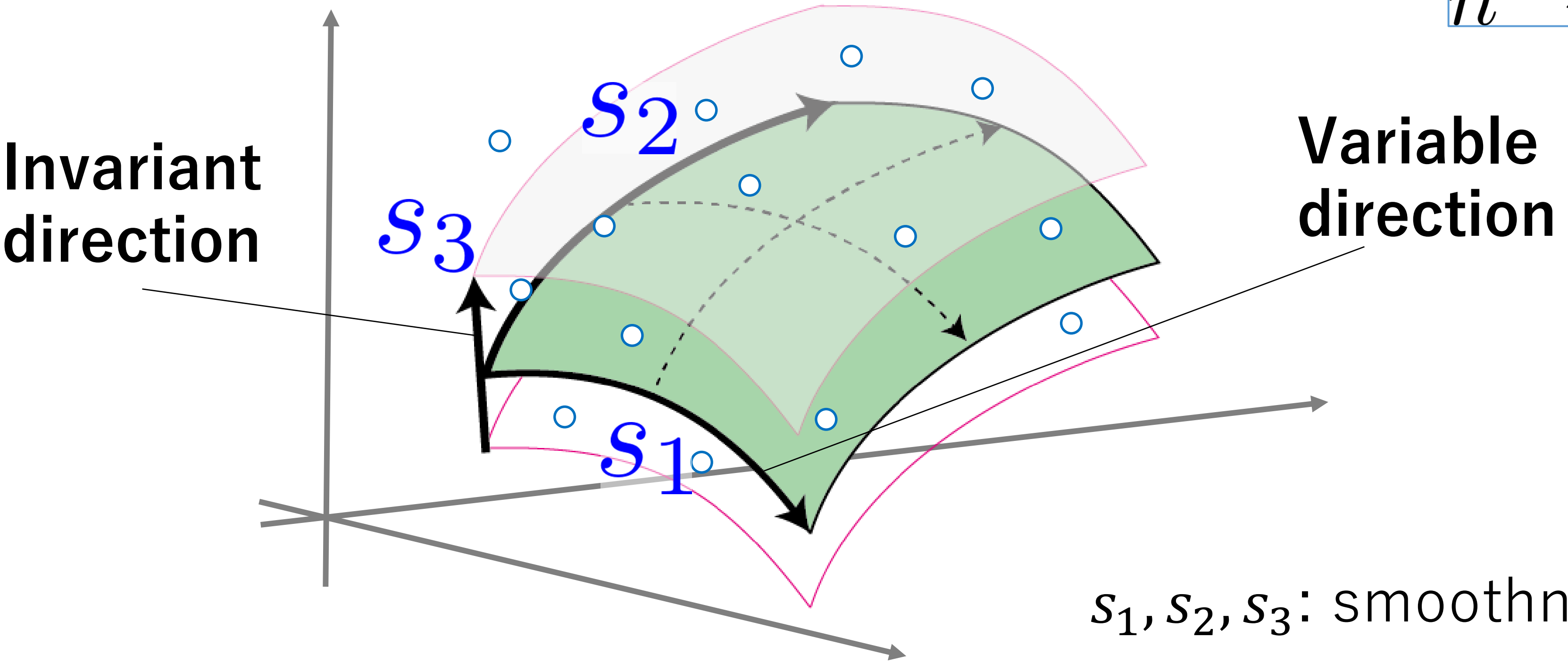
MNIST:
784 dim/13.4 intrinsic-dim
[Facco et al. 2017]

- **Classic nonparametric method:** Bickel & Li (2007); Yang & Tokdar (2015); Yang & Dunson (2016).
- **Deep learning:** Nakada & Imaizumi (2019); Schmidt-Hieber (2019); Chen et al. (2019).

$$n^{-\frac{2s}{2s+D}}$$

More realistic setting

$$n^{-\frac{2s}{2s+d}}$$



s_1, s_2, s_3: smoothness

(non-smooth) $s_1, s_2 \ll s_3$ (smooth)

Data are hardly distributed **exactly** on low-dim manifold.

- Smoothness could depend on directions.
- Local coordinate.

$$n^{-\frac{2\tilde{s}}{2\tilde{s}+1}} \qquad \tilde{s} = \left(s_1^{-1} + \cdots + s_d^{-1}\right)^{-1}$$

Anisotropic Besov space

Def. (Anisotropic Besov space)

$$\Delta_h^r(f)(x) := \Delta_h^{r-1}(f)(x+h) - \Delta_h^{r-1}(f)(x),$$
$$\Delta_h^0(f)(x) := f(x) \quad (h \in \mathbb{R}^d)$$

(finite difference)

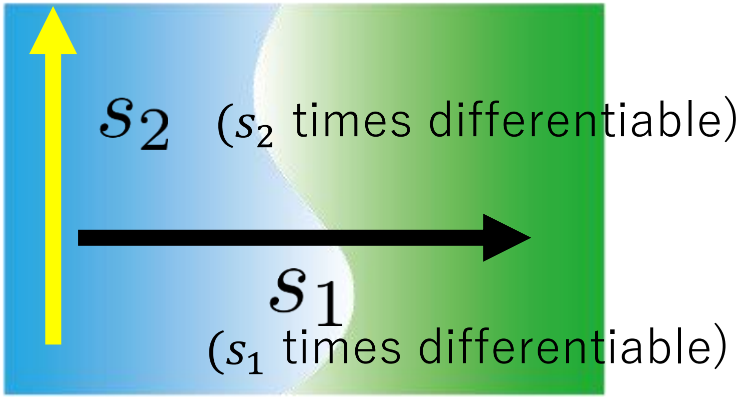
$$w_{r,p}(f,t) = \sup_{h\in\mathbb{R}^d : |h_i| \le t_i} \|\Delta_h^r(f)\|_p$$

(modulus of smoothness)

$$s = (s_1, \dots, s_d) \in \mathbb{R}_{++}^d$$

$$|f|_{B_{p,q}^s} := \begin{cases} \left(\sum_{k=0}^{\infty} [2^k w_{r,p}(f, (2^{-k/s_1}, \dots, 2^{-k/s_d}))]^q\right)^{1/q} & (q < \infty), \\ \sup_{k\ge 0} 2^k w_{r,p}(f, (2^{-k/s_1}, \dots, 2^{-k/s_d})) & (q = \infty). \end{cases}$$

$$\|f\|_{B_{p,q}^s} := \|f\|_p + |f|_{B_{p,q}^s}$$



L_p-norm of s_i-times derivative.

- s_i: smoothness to the i-th coordinate
- p: Uniformity of smoothness over the input space.

Anisotropic Besov space

Def. (Anisotropic Besov space)

$$\Delta_h^r(f)(x) := \Delta_h^{r-1}(f)(x+h) - \Delta_h^{r-1}(f)(x), \quad (h \in \mathbb{R}^d)$$
$$\Delta_h^0(f)(x) := f(x)$$

(finite difference)

$$w_{r,p}(f,t) = \sup_{h \in \mathbb{R}^d : |h_i| \le t_i} \|\Delta_h^r(f)\|_p$$

(modulus of smoothness)

$$s = (s_1, \dots, s_d) \in \mathbb{R}_{++}^d$$

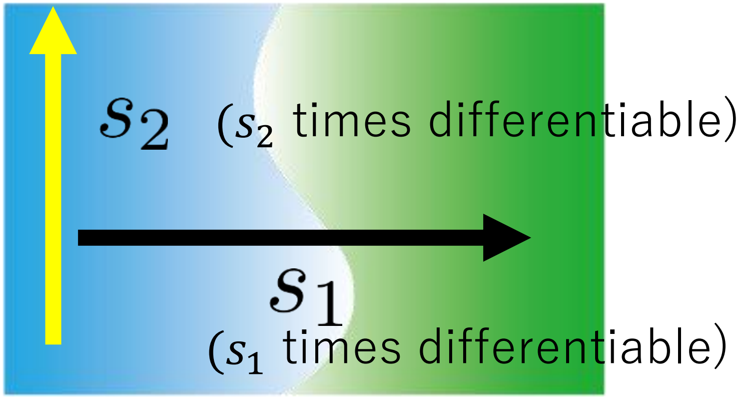
$$|f|_{B^s_{p,q}} := \begin{cases} \left(\sum_{k=0}^{\infty} [2^k w_{r,p}(f, (2^{-k/s_1}, \dots, 2^{-k/s_d}))]^q \right)^{1/q} & (q < \infty), \\ \sup_{k \ge 0} 2^k w_{r,p}(f, (2^{-k/s_1}, \dots, 2^{-k/s_d})) & (q = \infty). \end{cases}$$

$$\|f\|_{B^s_{p,q}} := \|f\|_p + |f|_{B^s_{p,q}}$$

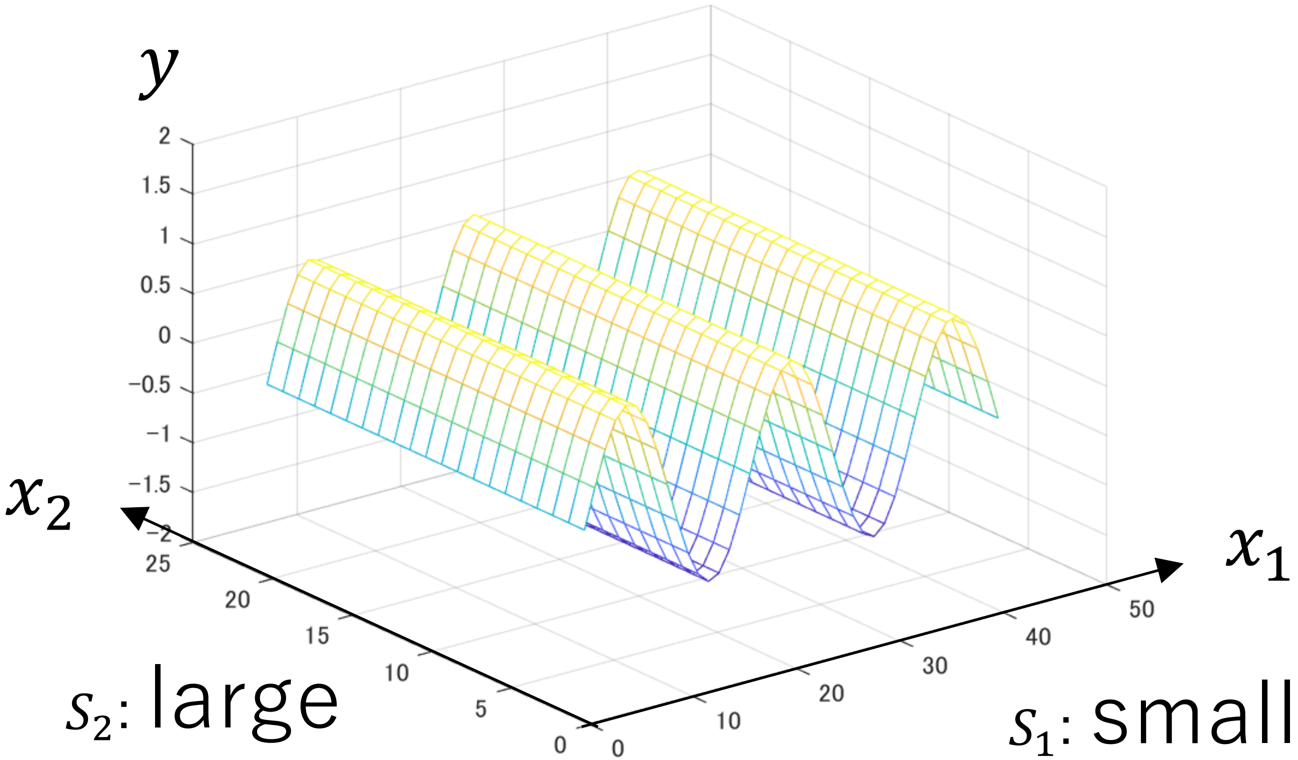
$$\|f\|_{B^s_{p,q}} = \|f\|_{L^p} + \sum_{i=1}^{d} \left\| \frac{\partial^{s_i} f}{\partial x_i^{s_i}} \right\|_{L^p}$$

L_p-norm of s_i-times derivative.

- s_i: smoothness to the i-th coordinate
- p: Uniformity of smoothness over the input space.



$$\|f\|_{B^{s}_{p,q}} = \|f\|_{L^{p}} + \sum_{i=1}^{d} \left\| \frac{\partial^{s_i} f}{\partial x_i^{s_i}} \right\|_{L^{p}}$$



Estimation error bound

$$f^\circ(x) = h_H \circ \cdots \circ h_1(x) \qquad h_\ell \in B_{p,q}^{(s_1^{(\ell)},\dots,s_{m_\ell}^{(\ell)})}([0,1]^{m_\ell}) \quad h_\ell : [0,1]^{m_\ell} \to [0,1]^{m_{\ell+1}}$$

$$\boxed{\hat f = \mathop{\arg\min}_{f:\ \text{deep neural-net}} \sum_{i=1}^{n} (y_i - f(x_i))^2}$$

(least squares estimator)

※ Here, we do not discuss optimization ability.

Theorem

Let $\tilde s^{(\ell)} := \left(\frac{1}{s_1^{(\ell)}} + \cdots + \frac{1}{s_{m_\ell}^{(\ell)}}\right)^{-1}$, $\quad \tilde s^{*(\ell)} := \tilde s^{(\ell)} \prod_{k=\ell+1}^{H} [(\min_j s_j^{(k)} - 1/p) \wedge 1]$

$$\boxed{\mathrm{E}[\|\hat f - f^\circ\|_{L^2(P_X)}^2] \lesssim \max_{\ell \in [H]} n^{-\frac{2\tilde s^{*(\ell)}}{2\tilde s^{*(\ell)}+1}} \log(n)^3}$$

The rate of convergence is determined by smoothness parameters.

When $H = 1$,

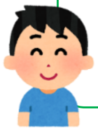
$$n^{-\frac{2\tilde s}{2\tilde s + 1}}$$

$$\tilde s = \left(s_1^{-1} + \cdots + s_d^{-1}\right)^{-1}$$

If s_is are small (non-smooth) toward small numbers of directions and large toward other directions, DNN can avoid the curse of dimensionality.

If $s_2 = \cdots = s_d = \infty$, then $n^{-\frac{2s_1}{2s_1+1}}$.

Comparison to linear estimator：



DNN

$$n^{-\frac{2\tilde s}{2\tilde s+1}}$$

$$\tilde s = \left(s_1^{-1} + s_2^{-1} + s_3^{-1}\right)^{-1}$$

Linear

$$n^{-\frac{2s_1}{2s_1+d}}$$

(Affected by dimension)

- **Linear estimator cannot find important directions. Then, the rate of convergence is strongly affected by the most non-smooth** (s_1) parameter**.**
- **We used a "convex hull argument" to show the rate of convergence.**

Comparison to linear estimator

$$f^\circ(x) = g(Wx) \qquad (W \in \mathbb{R}^{D\times d},\ g \in B^s_{p,q}([0,1]^D))$$

f° depends only D-dimensional subspace.

Deep

$$n^{-\frac{2s}{2s+D}}$$

($n^{-\frac{2s}{2s+1}}$ when $D = 1$)

Optimal

$\ll$

Linear estimator

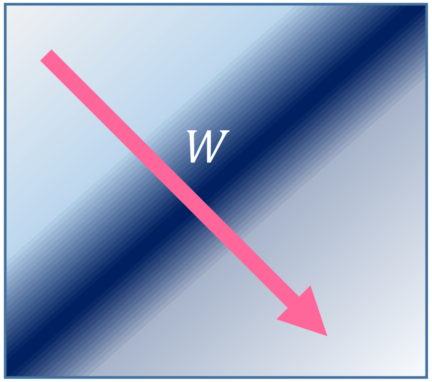
$$n^{-\frac{2(s-D/p+d/2+c)}{2(s-D/p+d/2+c)+d}}$$

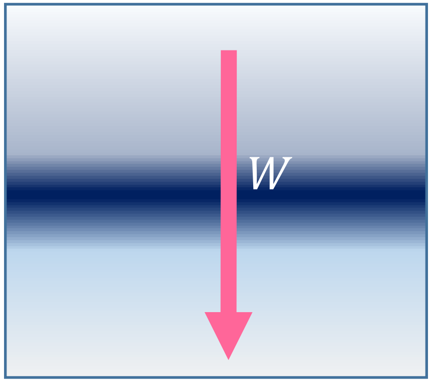
$c = 1$ if $D < d/2$, $c = 0$ if $D \geq d/2$.

($n^{-\frac{2s+d}{2s+2d}}$ when $D = 1$ and $p = 1$)

Suboptimal

Deep can ease curse of dim., but linear estimators directly suffers from curse of dim.





Outline

1. **Representation ability + Generalization ability**
   - Universal approximator
   - Depth separation
   - Adaptivity of deep learning
     - Inhomogeneity of smoothness
     - Curse of dimensionality
   - Foundation models
     - Diffusion model
     - Transformer
2. **Optimization ability**
   - Noisy gradient descent
   - Mean field Langevin
   - CSQ lowerbound

Analysis of diffusion model

- Kazusato Oko, Shunta Akiyama, Taiji Suzuki: Diffusion Models are Minimax Optimal Distribution Estimators. ICML2023.

Diffusion model

「An astronaut riding a horse in a photorealistic style」 ➡

DALL·E: [Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, Ilya Sutskever: Zero-Shot Text-to-Image Generation. ICML2021.]
DALL·E2:[Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, Mark Chen: Hierarchical Text-Conditional Image Generation with CLIP Latents. arXiv:2204.06125]

Stable diffusion, 2022.

Jason Allen "Théâtre D'opéra Spatial“ generated by **Midjourney**. Colorado State Fair’s fine art competition, 1st prize in digital art category

Generated by NovelAI

Movie generation

- SORA (OpenAI, 2024)
  - Diffusion Transformer

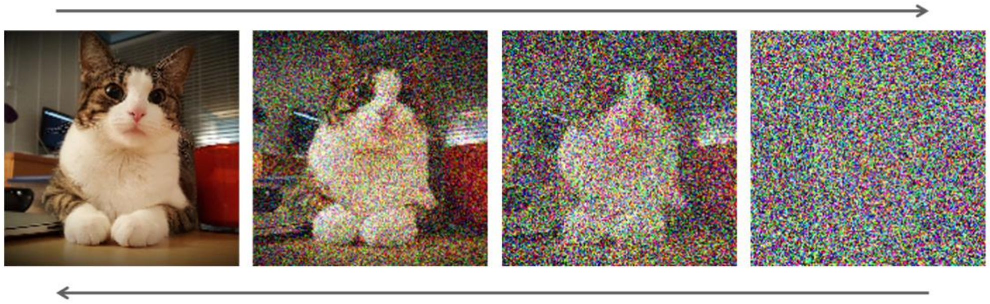
A stylish woman walks down a Tokyo street filled with warm glowing neon and animated city signage. She wears a black leather jacket, a long red dress, and black boots, and carries a black purse. She wears sunglasses and red lipstick. She walks confidently and casually. The street is damp and reflective, creating a mirror effect of the colorful lights. Many pedestrians walk about.

Diffusion model

[Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Song et al., 2020; Ho et al., 2020; Vahdat et al., 2021]

Forward process： Convert the target distribution to a noise distribution (e.g., Gaussian)

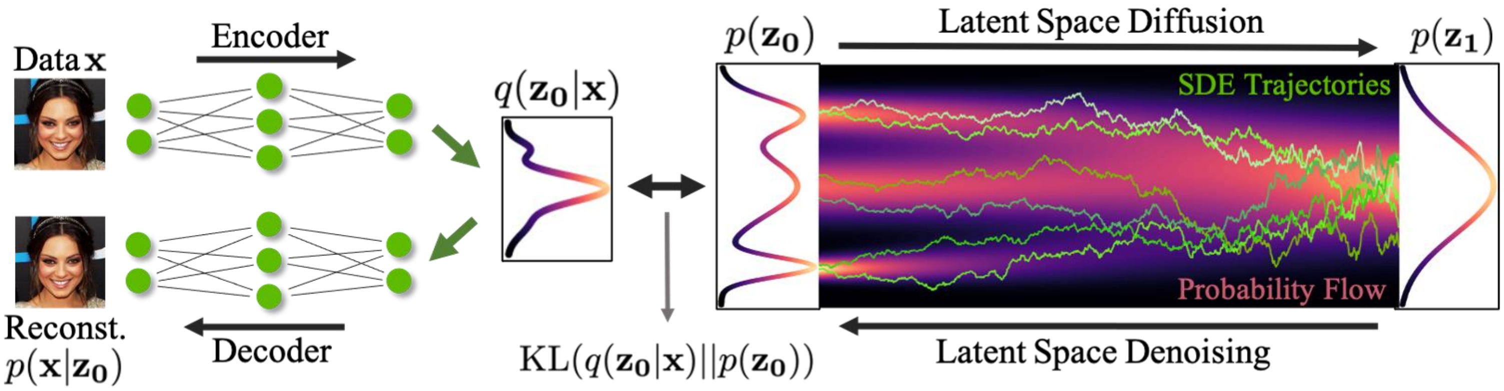
$$\mathrm{d}X_t = -X_t\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$



$$\mathrm{d}Y_t = (Y_t + 2\nabla \log(p_{\overline{T}-t}(Y_t))\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$

$(Y_t \sim X_{\overline{T}-t})$

Reverse process： Convert the noise distribution to the target distribution



[Vahdat, Kreis, Kautz: Score-based Generative Modeling in Latent Space. arXiv:2106.05931]

Forward process

Forward process: $(X_{t+\eta} = X_t - \eta X_t + \sqrt{2\eta}\xi_t)$

$$\mathrm{d}X_t = -X_t \mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$

OU process

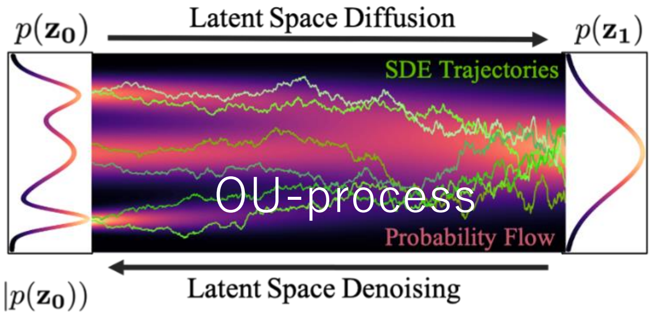
$$p_t = \mathrm{Law}(X_t) \Rightarrow p_t = \int N(\mu_t X_0, \sigma_t^2 I) p_0(\mathrm{d}X_0)$$

where $\mu_t = \exp(-t),\ \sigma_t^2 = 1 - \exp(-2t)$.

$$p_t(x) = \int p_0(y) \frac{1}{\sigma_t^d (2\pi)^{\frac{d}{2}}} \exp\left(-\frac{\|x - \mu_t y\|^2}{2\sigma_t^2}\right) \mathrm{d}y$$

The forward process converges to the noise distribution (standard normal) exponentially:

$$\mathrm{KL}(p_t || N(0, I)) \leq \exp(-2t) \mathrm{KL}(p_0 || N(0, I))$$



Standard normal

[Vahdat, Kreis, Kautz: Score-based Generative Modeling in Latent Space. arXiv:2106.05931]

Reverse process

Reverse process:

[Haussmann & Pardoux, 1986]

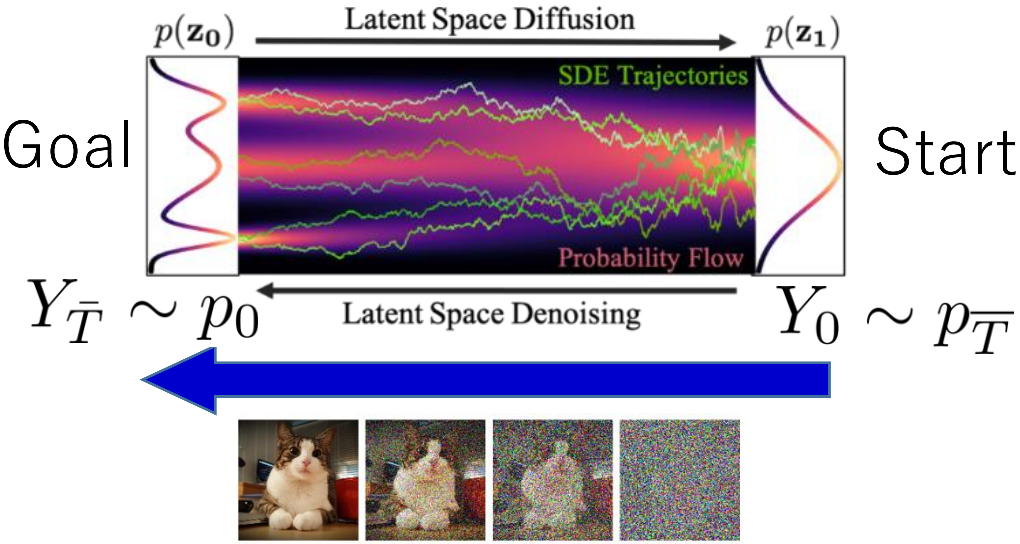
$$Y_0 \sim p_{\bar{T}}$$

$$\mathrm{d}Y_t = (Y_t + 2\nabla \log(p_{\bar{T}-t}(Y_t))\mathrm{d}t + \sqrt{2}\mathrm{d}B_t \qquad (t \in [0, \bar{T}])$$

Fact : Y_t's distribution$=X_{\bar{T}-t}$'s distribution

That is, $Y_t \sim p_{\bar{T}-t}$

By following the forward process in reverse, noise that follows a (nearly) normal distribution can be gradually modified to reproduce the original distribution of the images.



Goal

Start

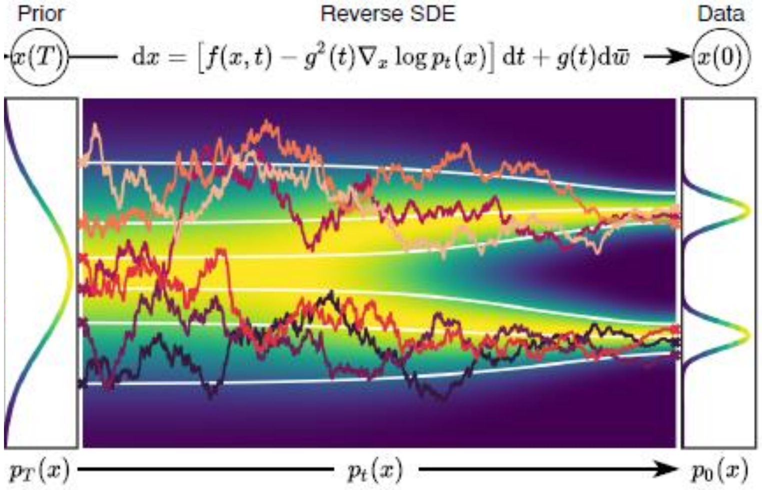
$Y_{\bar{T}} \sim p_0$

$Y_0 \sim p_{\bar{T}}$

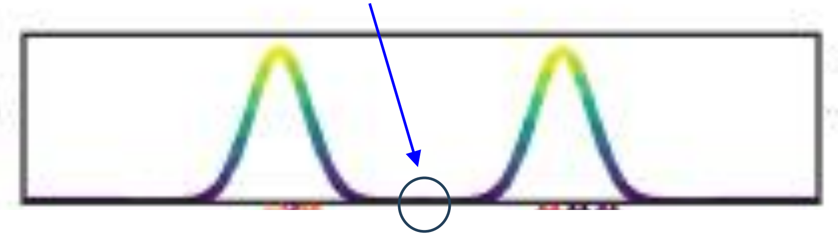
Benefit of diffusion model

We can sample from multimodal distribution efficiently.

- ➢“Easy distribution” →”Difficult distribution”



If we try to sample directly from the original distribution, then it could not get over the “gap”.



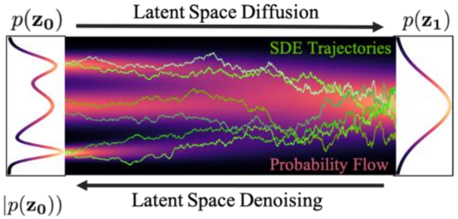
- Even though the score of the original distribution is complex, the distribution of the diffused X_t is smooth → easy to estimate → easy to generalize.
- Learning is more stable because it uses information of the intermediate distribution p_t instead of directly learning end-to-end mapping from the noise to the source distribution.

Toy data

[https://github.com/Kei18/tiny-tiny-diffusion]

This is not generating an image of a dinosaur but the shape of the density function looks like a dinosaur. Each point corresponds to each image (X_t).

Score estimation



Reverse process:

(unknown)

$$Y_0 \sim p_{\overline{T}}$$

(unknown)

$$\mathrm{d}Y_t = (Y_t + 2\underline{\nabla \log(p_{\overline{T}-t}(Y_t))})\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$

[Haussmann & Pardoux, 1986]

$$\Rightarrow Y_t \sim p_{\overline{T}-t}$$

$$\boldsymbol{Y_{\overline{T}} \sim p_0}$$

Approximated process (generative model):

$$\hat{Y}_0 \sim N(0, I)$$

($N(0,I)$ is close to $p_{\overline{T}}$)

$$\mathrm{d}\hat{Y}_t = (\hat{Y}_t + 2\hat{s}(\hat{Y}_t, \overline{T} - t))\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$

Theorem (Girsanov's theorem)

If $\hat{Y}_0 \sim p_{\overline{T}}$, then

$$\mathrm{KL}(p_0 || p_{\hat{Y}_{\overline{T}}}) \le \frac{1}{4}\int_0^{\overline{T}} \mathbb{E}_{Y_t}[\|\nabla \log(p_{\overline{T}-t}(Y_t)) - \hat{s}(Y_t, \overline{T} - t)\|^2]\mathrm{d}t$$

⇒ It suffices to estimate the *score function* $\nabla \log(p_t)$ as accurate as possible.

Score matching

Reference

$$\int_0^{\overline{T}} \mathbb{E}_{Y_t}[\|\underline{\nabla \log(p_{\overline{T}-t}(Y_t))} - \hat{s}(Y_t, \overline{T}-t)\|^2]\mathrm{d}t$$

Unknown. We cannot calculate.

$$= \int_0^{\overline{T}} \mathbb{E}_{X_t}[\|\nabla \log(p_t(X_t)) - \hat{s}(X_t, t)\|^2]\mathrm{d}t$$

（$X_{\overline{T}-t}$とY_tは同じ分布）

$$= \int_0^{\overline{T}} \mathbb{E}_{X_t}[\|\nabla \log(p_t(X_t))\|^2 - 2\langle \nabla \log(p_t(X_t)), \hat{s}(X_t, t)\rangle + \|\hat{s}(X_t, t)\|^2]\mathrm{d}t$$

$$= \int_0^{\overline{T}} \mathbb{E}_{X_t}\left[\underbrace{-2\left\langle \frac{\nabla p_t(X_t)}{p_t(X_t)}, \hat{s}(X_t, t)\right\rangle + \|\hat{s}(X_t, t)\|^2}\right]\mathrm{d}t + (\mathrm{const})$$

$$\int -2\left\langle \frac{\nabla p_t(x_t)}{\cancel{p_t(x_t)}}, \hat{s}(x_t, t)\right\rangle \cancel{p_x(x_t)}\mathrm{d}x_t + \mathbb{E}\left[\|\hat{s}(X_t, t)\|^2\right]$$

$$= \int -2\left\langle \nabla p_t(x_t), \hat{s}(x_t, t)\right\rangle \mathrm{d}x_t + \mathbb{E}\left[\|\hat{s}(X_t, t)\|^2\right]$$

$$= \int -2\left\langle \nabla \int p_t(x_t|x_0)p_0(x_0)\mathrm{d}x_0, \hat{s}(x_t, t)\right\rangle \mathrm{d}x_t + \mathbb{E}\left[\|\hat{s}(X_t, t)\|^2\right]$$

$$= \int\int -2\left\langle \nabla p_t(x_t|x_0), \hat{s}(x_t, t)\right\rangle p_0(x_0)\mathrm{d}x_0\mathrm{d}x_t + \mathbb{E}\left[\|\hat{s}(X_t, t)\|^2\right]$$

$$= \int\int -2\left\langle \frac{\nabla p_t(x_t|x_0)}{\underline{p_t(x_t|x_0)}}, \hat{s}(x_t, t)\right\rangle \underline{p_t(x_t|x_0)}p_0(x_0)\mathrm{d}x_0\mathrm{d}x_t + \mathbb{E}\left[\|\hat{s}(X_t, t)\|^2\right]$$

$$= \mathbb{E}_{X_0, X_t}\left[-2\left\langle \nabla \log(p_t(X_t|X_0)), \hat{s}(X_t, t)\right\rangle + \|\hat{s}(X_t, t)\|^2\right]$$

Score matching

Reference

$$\int_0^{\overline{T}} \mathbb{E}_{Y_t}[\|\nabla \log(p_{\overline{T}-t}(Y_t)) - \hat{s}(Y_t, \overline{T}-t)\|^2]\mathrm{d}t$$

$$= \int_0^{\overline{T}} \mathbb{E}_{X_t}[\|\nabla \log(p_t(X_t)) - \hat{s}(X_t, t)\|^2]\mathrm{d}t$$

$$= \int_0^{\overline{T}} \mathbb{E}_{X_t}[\|\nabla \log(p_t(X_t))\|^2 - 2\langle \nabla \log(p_t(X_t)), \hat{s}(X_t, t)\rangle + \|\hat{s}(X_t, t)\|^2]\mathrm{d}t$$

$$= \int_0^{\overline{T}} \mathbb{E}_{X_t} \underbrace{\left[-2\left\langle \frac{\nabla p_t(X_t)}{p_t(X_t)}, \hat{s}(X_t, t)\right\rangle + \|\hat{s}(X_t, t)\|^2\right]}_{\mathbb{E}_{X_0, X_t}\left[-2\left\langle \nabla \log(p_t(X_t|X_0)), \hat{s}(X_t, t)\right\rangle + \|\hat{s}(X_t, t)\|^2\right]} \mathrm{d}t + (\mathrm{const})$$

$$= \int_0^{\overline{T}} \mathbb{E}_{X_t, X_0}[\|\nabla \log(p_t(X_t|X_0)) - \hat{s}(Y_t, t)\|^2]\mathrm{d}t + (\mathrm{const})$$

Score matching

$$\int_0^{\overline{T}} \mathbb{E}_{Y_t}[\|\nabla \log(p_{\overline{T}-t}(Y_t)) - \hat{s}(Y_t, \overline{T}-t)\|^2]\mathrm{d}t$$

$$= \int_0^{\overline{T}} \mathbb{E}_{X_t}[\|\nabla \log(p_t(X_t)) - \hat{s}(X_t, t)\|^2]\mathrm{d}t$$

$$= \int_0^{\overline{T}} \mathbb{E}_{X_t, X_0}[\|\nabla \log(p_t(X_t|X_0)) - \hat{s}(Y_t, t)\|^2]\mathrm{d}t + (\mathrm{const})$$

Observation (n data points $D_n = \{x_i\}_{i=1}^n$):

$$x_i \sim p_0 \quad (i = 1, \ldots, n)$$

Empirical score matching loss:

$$\min_{s \in \mathrm{DNN}} \frac{1}{n} \sum_{i=1}^n \int_{t=\underline{T}}^{\overline{T}} \mathbb{E}_{X_t|X_0=x_i}\left[\|s(X_t, t) - \underline{\nabla \log p_t(X_t|x_i)}\|^2\right] \mathrm{d}t$$

Can be sampled via OU process

$$N(x_i e^{-t}, 1 - e^{-2t})$$

Explicit form is available

$$-\frac{(X_t - e^{-t} x_i)}{1 - e^{-2t}}$$

Error analysis of diffusion models

- Reverse SDE characterization: Song et al. (2021)

[Approximation error analysis]

- KL-divergence bound via Girsanov's theorem: Chen et al. (2022)

- Error bound with LSI: Lee et al. (2022a)
    - With smoothness: Chen et al. (2022) and Lee et al. (2022b)

- Error propagation with manifold assumption: Pidstrigach (2022)

[Generalization analysis]

- Wasserstein distance bound: $O(n^{-\frac{1}{d}})$ with manifold assumption: De Bortoli (2022)

Q:

1. How accurately can we estimate the score functions?
2. How strongly does the estimation error of score functions affect the final result?

Problem setting

Assumption 1

The true distribution p_0 is supported on $[-1,1]^d$ and

$$p_0 \in B^s_{p,q}$$

with $s > (1/p - 1/2)_+$ as a density function on $[-1,1]^d$.

Assumption2

p_0 is sufficiently smooth on the edge of the support $[-1,1]^d \setminus [-1+n^{-\frac{1-\delta}{d}}, 1-n^{-\frac{1-\delta}{d}}]^d$.

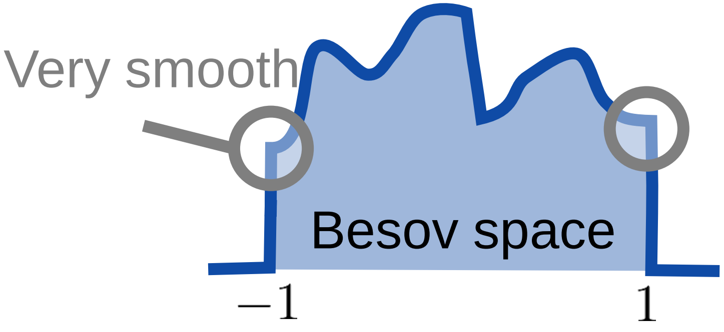
Besov space $(B^s_{p,q}(\Omega))$

$$\omega_m(f,t)_p := \sup_{\|h\| \le t} \left\| \sum_{j=1}^{m} (-1)^{m-j} \binom{m}{j} f(\cdot + jh) \right\|_{L^p(\Omega)},$$

$$\|f\|_{B^s_{p,q}(\Omega)} = \|f\|_{L^p(\Omega)} + \left(\int_0^\infty [t^{-s} \omega_m(f,t)_p]^q \frac{\mathrm{d}t}{t} \right)^{1/q}.$$

Smoothness

Spatial inhomogeneity



Problem setting

Assumption 1

The true distribution p_0 is supported on $[-1,1]^d$ and

$$p_0 \in B^s_{p,q}$$

with $s > (1/p - 1/2)_+$ as a density function on $[-1,1]^d$.

Assumption2

p_0 is sufficiently smooth on the edge of the support $[-1,1]^d \setminus [-1+n^{-\frac{1-\delta}{d}}, 1-n^{-\frac{1-\delta}{d}}]^d$.

Besov space $(B^s_{p,q}(\Omega))$

Intuition

$$\|f\|_{B^s_{p,q}(\Omega)} = \|f\|_{L^p(\Omega)} + \|D^s f\|_{L^p(\Omega)}$$

Smoothness

Uniformity of smoothness

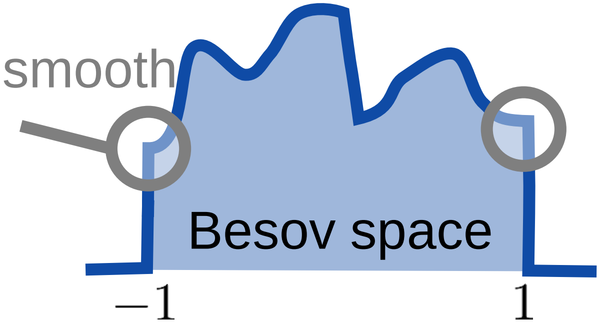
$\omega_m(f,t)_p$

$$\|f\|_{B^s_{p,q}(\Omega)} = \|f\|_{L^p(\Omega)} + \left(\int_0^\infty [t^{-s}\omega_m(f,t)_p]^q \frac{dt}{t}\right)^{1/q}.$$

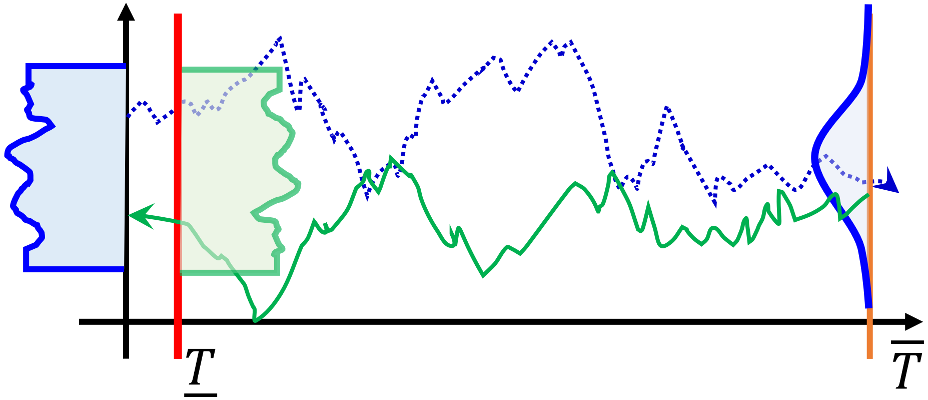
Smoothness

Spatial inhomogeneity

ery smooth



Convergence rate result



Theorem (Estimation error in TV-distance)

Let $\underline{T} = n^{-O(1)}, \overline{T} = O(\log(n))$. Then, the empirical risk minimizer $\hat{s}$ in DNN satisfies

$$\mathbb{E}_{D_n}\left[\mathrm{TV}(\hat{Y}_{\overline{T}-\underline{T}}, X_0)\right] \lesssim n^{-\frac{s}{2s+d}} \log^9(n).$$

This is minimax optimal, that is, it holds

$$n^{-\frac{s}{2s+d}} \lesssim \inf_{\hat{\mu}:\text{estimator}} \sup_{p_0} \mathbb{E}_{D_n}\left[\mathrm{TV}(\hat{\mu}, X_0)\right]$$

Although $\hat{s}(x,t)$ is a function with $d+1$-dimensional input, there appears “d” in the bound instead of $d+1$. This is because Gaussian convolution induces smoothness.

B-spline basis decomposition

Reference

$$\nabla \log(p_t(x)) = \frac{\boxed{\nabla p_t(x)}}{\boxed{p_t(x)}}$$

Approximate each term by DNNs

- B-spline decomposition of a Besov function p_0

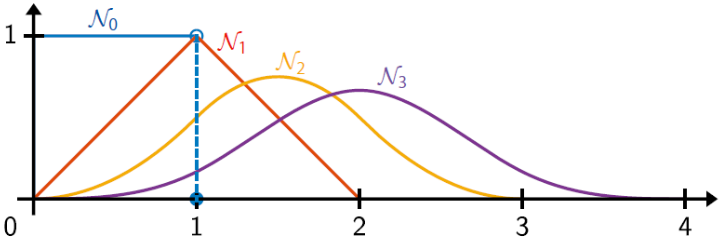
$$p_0(x) \approx \sum_{j=1}^{N} \alpha_j M^d_{a^j,b^j}(x)$$

$$\mathcal{N}(x) = \begin{cases} 1 & (x \in [0,1]), \\ 0 & (\text{otherwise}) \end{cases}$$

Cardinal B-spline of order m:

$$\mathcal{N}_m(x) = (\underbrace{\mathcal{N} * \mathcal{N} * \cdots * \mathcal{N}}_{m+1 \text{ times}})(x)$$

→ Piece-wise polynomial of order m.



Tensor product B-spline:

$$M^d_{a,b}(x) = \prod_{j=1}^{d} \mathcal{N}_m(2^{a_j} - b_j)$$

Cardinal B-spline interpolation (DeVore & Popov, 1988)

Reference

- Atomic decomposition:

$$\mathcal{N}^{(d)}_{k,j}(x_1,\dots,x_d)=\prod_{i=1}^{d}\mathcal{N}_m(2^k x_i - j_i)$$

$f \in B^s_{p,q}$ can be decomposed into

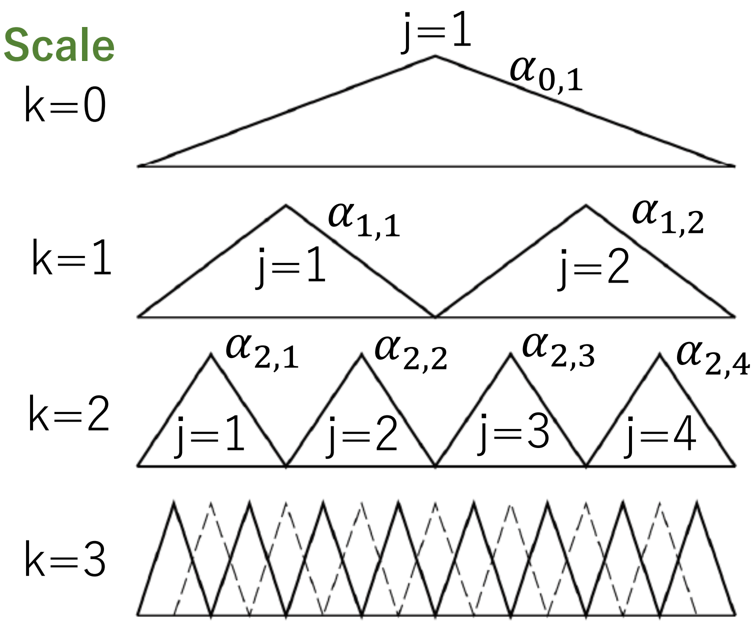
$$f=\sum_{k\in\mathbb{N}+}\sum_{j\in J(k)}\alpha_{k,j}\mathcal{N}^{(d)}_{k,j}$$

such that
(where $J(k)=\{j\in\mathbb{Z}^d \mid -m<j_i<2^{k_i+1}+m\}$)

$$N(f)=\left[\sum_{k=0}^{\infty}\{2^{sk}(2^{-kd}\sum_{j\in J(k)}|\alpha_{k,j}|^p)^{1/p}\}^q\right]^{1/q}<\infty$$

$\|f\|_{B^s_{p,q}} \simeq N(f)$ (Norm equivalence)

Wavelet/multi-resolution expansion



DNN can approximate each B-spline basis efficiently.

$$f=\underline{\sum_{k,j\in I_N}\alpha_{k,j}\mathcal{N}^{(d)}_{k,j}}+O(N^{-s/d})$$

N terms (should be appropriately chosen depending on f)

(see also Bolcskei, Grohs, Kutyniok, Petersen: Optimal Approximation with Sparsely Connected Deep Neural Networks. 2018)

Proof outline (1)

$$\nabla \log(p_t(x)) = \frac{\boxed{\nabla p_t(x)}}{\boxed{p_t(x)}}$$

Approximate each term by DNNs

- B-spline decomposition of a Besov function p_0

$$p_0(x) \approx \sum_{j=1}^{N} \alpha_j M^d_{a^j,b^j}(x)$$

Approximation error $O(N^{-s/d})$

- Diffused B-spline basis expansion of p_t

$$p_t(x) = \int \boxed{p_0(y)} \underbrace{\frac{1}{\sigma_t^d (2\pi)^{\frac{d}{2}}} \exp\left(-\frac{\|x-\mu_t y\|^2}{2\sigma_t^2}\right)}_{=: K_t(x|y)} \mathrm{d}y$$

Decompose

$$p_t(x) \approx \sum_{j=1}^{N} \alpha_j \underbrace{\int M^d_{a^j,b^j}(y) K_t(x|y) \mathrm{d}y}_{=: E_{a^j,b^j}(x,t)}$$

Diffused B-spline

➢ We approximate *Diffused B-splines* by DNNs.

Approximation error of Diffused B-spline

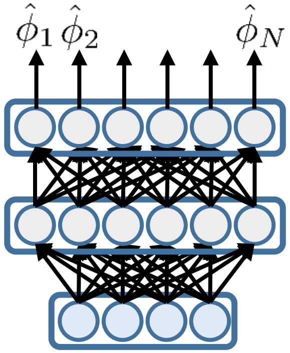
Lemma (Approximation error of diffused B-spline)

There exists a deep neural network $\hat{\phi}: \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^d$ such that

$$\left\| \hat{\phi}(x,t) - E_{a^j,b^j}(x,t) \right\|_\infty \leq \epsilon$$

with depth $L = O\left(\log^4(\epsilon^{-1})\right)$, width $W_i = O(\log^6(\epsilon^{-1}))$, sparsity (# of non-zero parameters) $S = O(\log(\epsilon^{-1}))$, and ℓ^∞-norm bound $B = O(\exp(O(\log^2(\epsilon^{-1}))))$ on parameters.

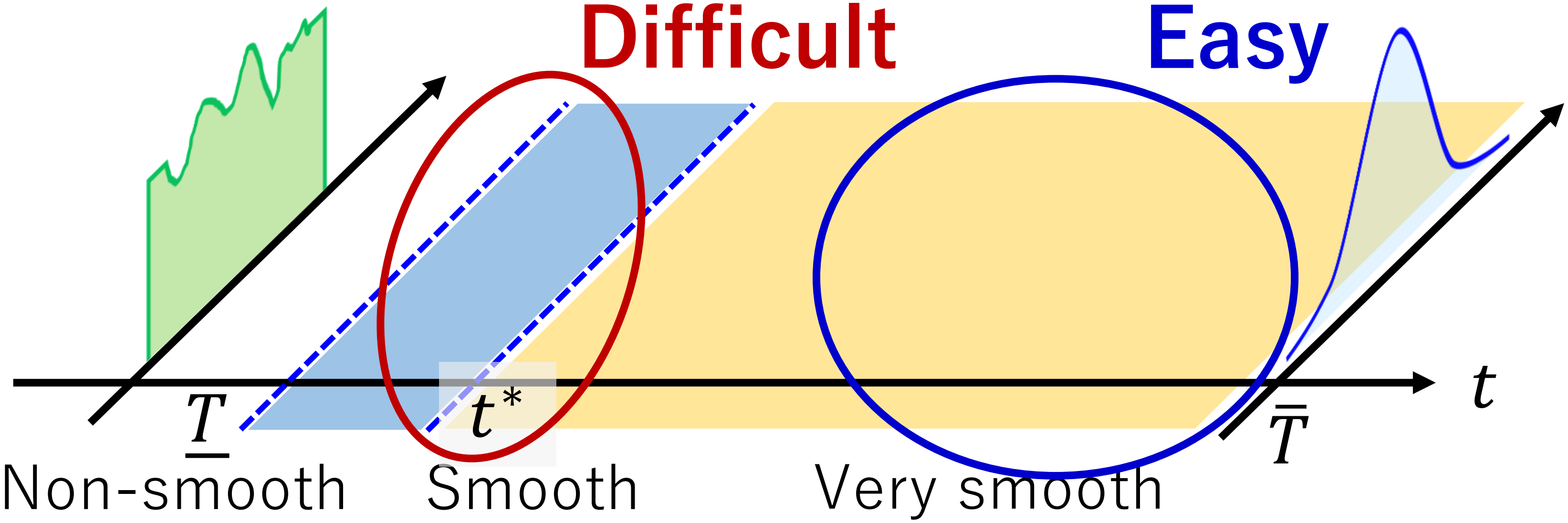
$$\check{f}_N(x,t) = \sum_{i=1}^{N} \alpha_i \hat{\phi}_i(x,t)$$: Deep neural network



# of non-zero parameters: $N \mathrm{polylog}(N)$

$$\|p_t(\cdot) - \check{f}_N(\cdot,t)\|_{L^r} \leq \sum_{i=1}^N |\alpha_i| \underbrace{\|\phi_i(\cdot,t) - \hat{\phi}_i(\cdot,t)\|_{L^r}}_{\leq O(e^{-L})} + \underbrace{\|\textstyle\sum_{i=N+1}^\infty \alpha_i \phi_i(\cdot,t)\|_{L^r}}_{\leq N^{-s/d}}$$

Error bound of score



- Bound by diffused B-spline approximation

$$\|p_t - \check{f}_N(\cdot,t)\|_{L^r(X_t)} \lesssim N^{-s/d}\|p_0\|_{B^s_{p,q}}$$

$$\boxed{p_t(x) \approx \sum_{j=1}^{N} \alpha_j E_{a^j,b^j}(x,t)}$$

  - Similar argument is applied to ∇p_t:

$$\|\nabla \log p_t - \dot{f}_N(\cdot,t)\|_{L^2}^2 \lesssim \frac{N^{-2s/d}\log(N)}{\sigma_t^2}$$

$(\sigma_t^2 = 1 - \exp(-2t))$

- A tighter bound on the smooth part $(t > t_*)$

$$\|p_t\|_{W_p^k} = \sum_{|\alpha|\le k} \left\|\frac{\partial^\alpha p_t}{\partial x^\alpha}\right\|_{L^p} \lesssim \sigma_t^{-k} (\le t_*^{-\frac{k}{2}})$$

$$\Rightarrow \|p_t - \check{f}_{N'}\|_{L^2(X_t)}^2 \lesssim N'^{-2k/d} t_*^{-k}$$

(take $k = s + 1$)

- Useful for W1 bound.
- Smoothness around the edge (A2) is not requires.

Error decomposition

$\mathrm{TV}(X_0,\hat{Y}_{\overline{T}-\underline{T}}) \leq \sqrt{\frac{1}{2}\mathrm{KL}(X_0||\hat{Y}_{\overline{T}-\underline{T}})}$ (Pinsker's inequality)

Score matching loss

$$\mathrm{TV}(X_0,\hat{Y}_{\overline{T}-\underline{T}}) \lesssim \left[\int_{t=\underline{T}}^{\overline{T}} \mathbb{E}_{X_t\sim p_t}[\|\hat{s}(X_t,t) - \nabla \log p_t(X_t)\|^2]\mathrm{d}t\right]^{\frac{1}{2}}$$

$$+ n^{O(1)}\sqrt{\underline{T}} + \exp(-O(\overline{T})) \lesssim n^{-\frac{s}{d+2s}} \log^9 n$$

Truncation loss at $\underline{T}$.

Truncation loss at $\overline{T}$.

$$t_* = N^{-(2-\delta)/d}$$

$$\frac{\log(\text{covering num})}{n}$$

$$\int_{t=\underline{T}}^{\overline{T}} \mathbb{E}_{X_t}[\|\nabla \log p_t - \hat{s}(\cdot,t)\|^2]\mathrm{d}t$$

Bias

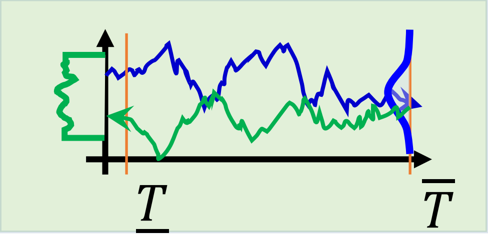
Variance

$$\lesssim \int_{\underline{T}}^{\overline{T}} \frac{N^{-2s/d}}{\sigma_t^2}\log(N)\mathrm{d}t + \frac{N\mathrm{polylog}(N)}{n}$$

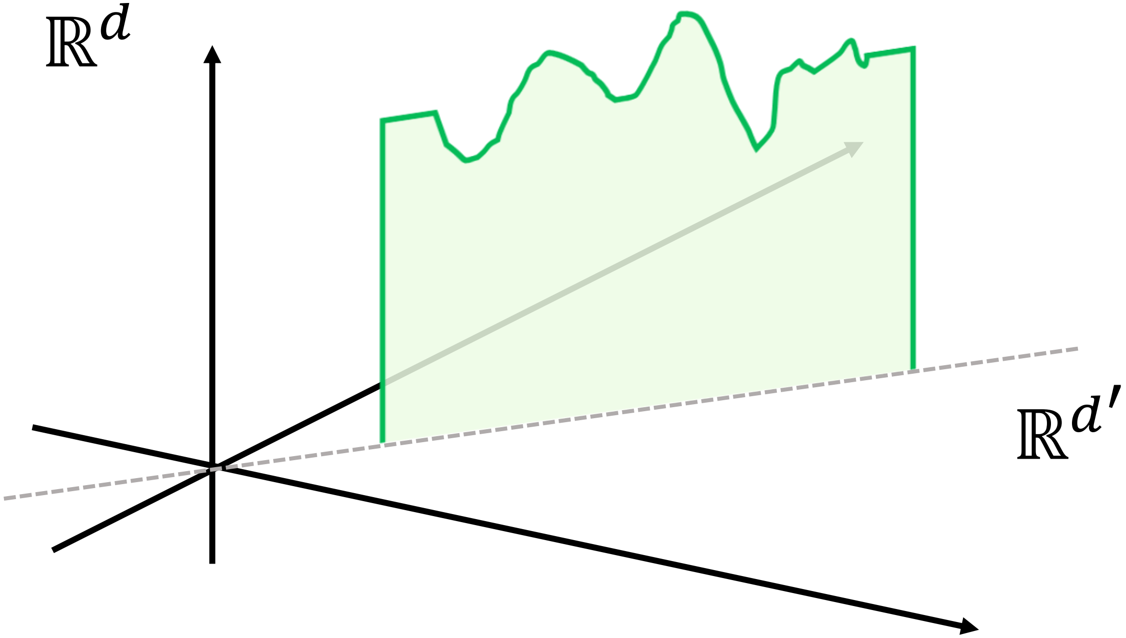
$$\lesssim \left(N^{-2s/d} + \frac{N}{n}\right)\mathrm{polylog}(N)$$

$N \simeq n^{d/(2s+d)}$

$$\lesssim n^{-2s/(2s+d)}\mathrm{polylog}(n)$$



Low dimensional structure



The support of the target distribution is in a low dimensional subspace.

The estimated distribution is never absolutely continuous to the target distribution.
→ **Wasserstein distance**

W_1-distance convergence rate

Theorem (Estimation error in W1-distance)

For any fixed $\delta > 0$, by slightly changing the estimator, the empirical risk minimizer $\hat{s}$ in DNN satisfies

$$\mathbb{E}_{D_n}\left[\mathrm{W}_1(\hat{Y}_{\overline{T}-\underline{T}}, X_0)\right] \lesssim n^{-\frac{s+1-\delta}{2s+d'}}.$$

This is also known as minimax optimal (up to δ) [Niles-Weed & Berthet (2022)].

- d' appears instead of d: **Diffusion model can avoid curse of dimensionality**.
- The minimax rate of Wasserstein distance is faster than that of TV distance, which makes it difficult to establish the bound.
  - ➢ We need more precise estimate of the score around $t = 0$.

(TV) $n^{-\frac{s}{2s+d}}$ ➡ $n^{-\frac{s+1}{2s+d}}$ (W1)

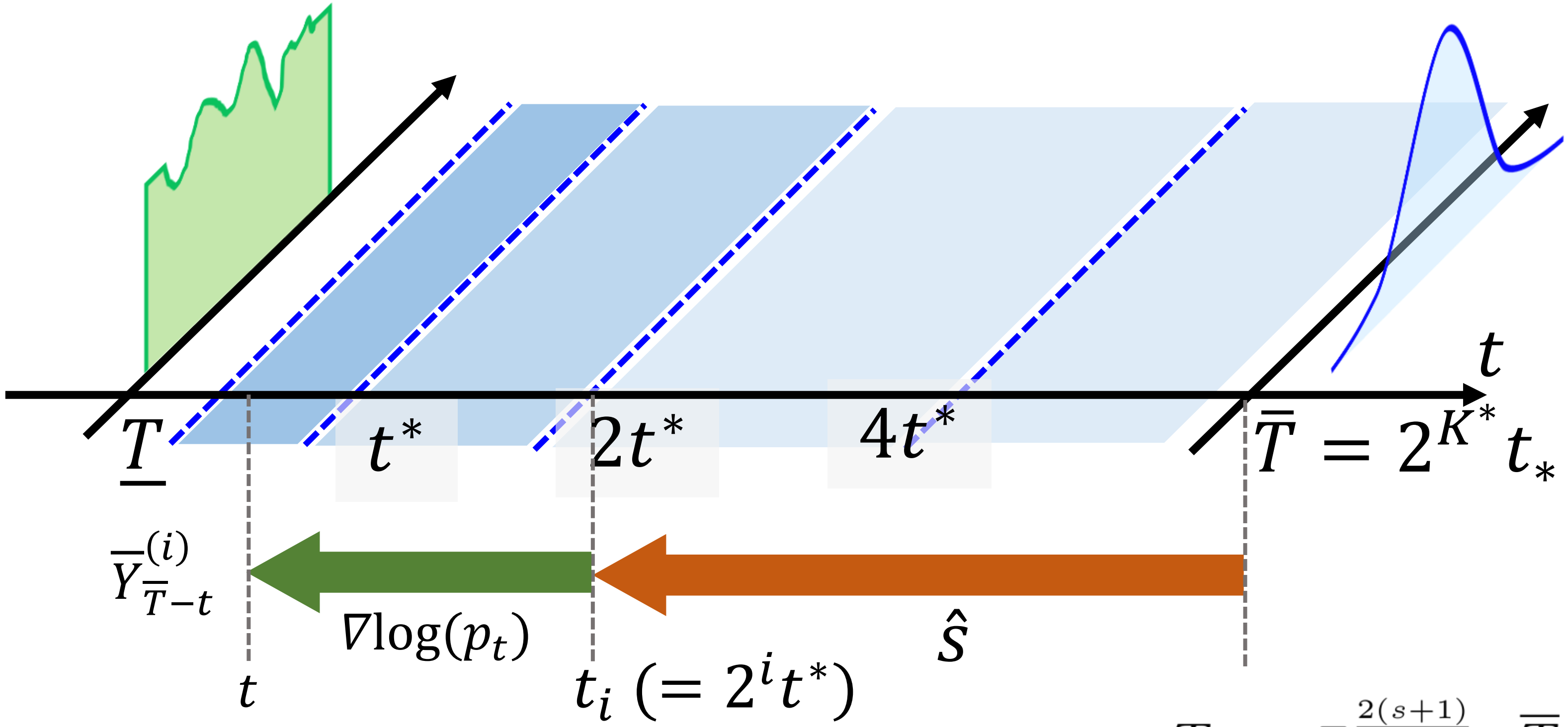
Lemma (tighter bound on W1 distance error)

$$\mathrm{W}_1(X_0, \hat{Y}_{\overline{T}-\underline{T}}) \lesssim \sqrt{\int_{t=\underline{T}}^{\overline{T}} t \, \mathbb{E}_{X_t}[\|\hat{s}(X_t, t) - \nabla \log p_t(X_t)\|^2] \mathrm{d}t} + \sqrt{\underline{T}} + \exp(-O(\overline{T}))$$

- For large t, we can estimate the score more accurately.
- For small t, the error does not propagate so much due to the term t.

→ **Better rate**.

Bound for W1 distance



$$\underline{T} = n^{-\frac{2(s+1)}{2s+d}}, \; \overline{T} \simeq \log(n)$$

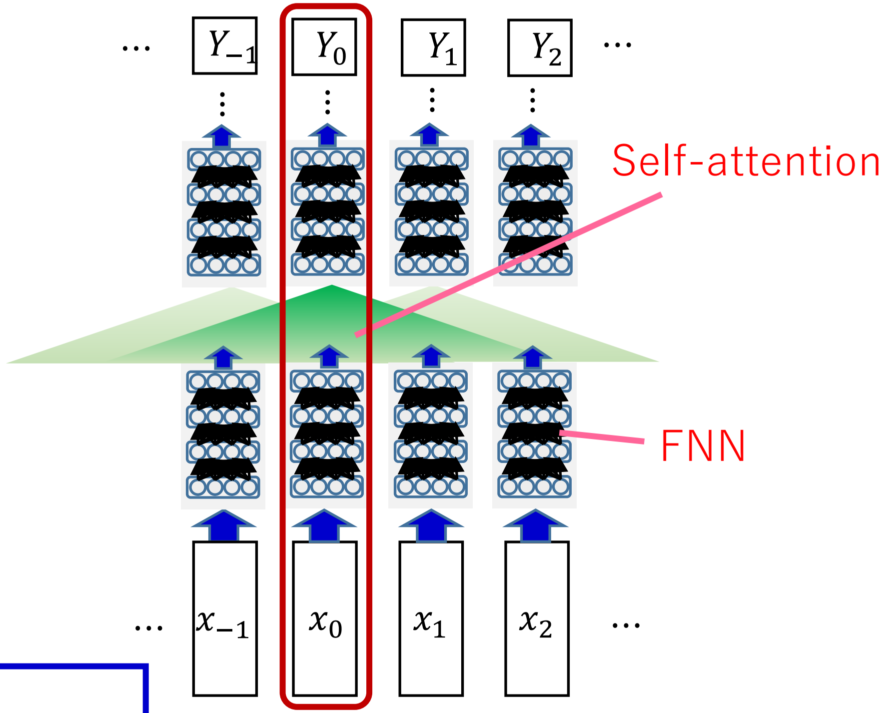
$$t_* = n^{-\frac{2-\delta}{d+2s}}, \quad t_k = t_* 2^k$$

$$\mathrm{W}_1(X_0, \hat{Y}_{\overline{T}-\underline{T}}) \le \underset{\text{(negligible)}}{\mathrm{W}_1(X_0, X_{\underline{T}})} + \underset{(\exp(-\overline{T}))}{\mathrm{W}_1(X_{\underline{T}}, \bar{Y}^{(K^*)}_{\overline{T}-\underline{T}})} + \sum_{i=1}^{K^*} \underbrace{\mathrm{W}_1(\bar{Y}^{(i-1)}_{\overline{T}-\underline{T}}, \bar{Y}^{(i)}_{\overline{T}-\underline{T}})}$$

$$\sqrt{t_{i-1} \int_{t_{i-1}}^{t_i} \mathbb{E}_{X_t}[\|\hat{s}(X_t,t) - \nabla \log p_t(X_t)\|^2] \mathrm{d}t} \lesssim n^{-\frac{s+1-\delta}{2s+d}}$$

Transformer

[Shokichi Takakura, Taiji Suzuki: Approximation and Estimation Ability of Transformers for Sequence-to-Sequence Functions with Infinite Dimensional Input. ICML2023]



Properties of Transformer

- It can output a value from wide rage of tokens.

→ Curse of dimensionality?

- It can choose important tokens depending on input.

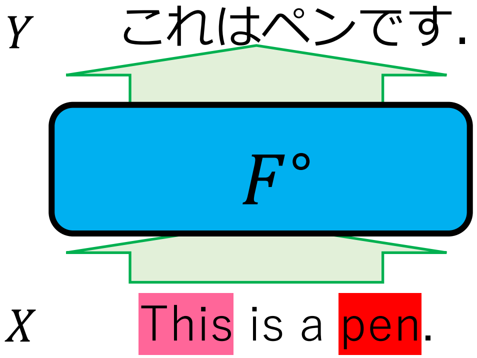
→ Can avoid curse of dim!

We showed minimax optimality to estimate a sequence-to-sequence function.

Theorem（estimation error）

$$\frac{1}{r-l+1}\sum_{j=l}^{r}\mathbb{E}[\|\hat{F}_j - F_j^\circ\|_{L_2(P_X)}^2] \lesssim n^{-\frac{2a^\dagger}{2a^\dagger+1}}(\log n)^{2/\alpha+2+\max\{4/\alpha,4\}}$$

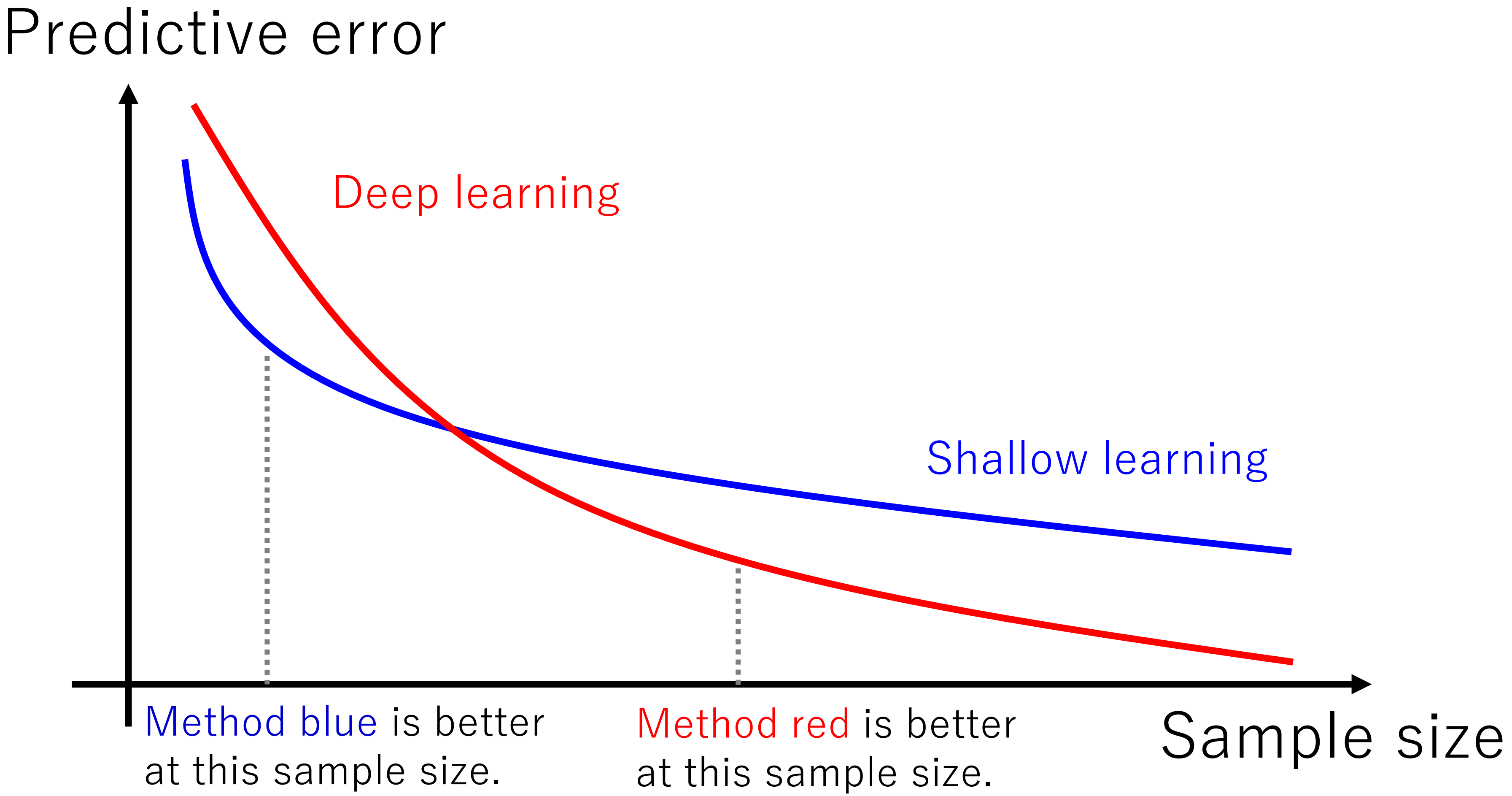
(almost minimax optimal)



- ➢ It achieves polynomial order convergence even though input is infinite-dimensional.

Remark on the rate of convergence

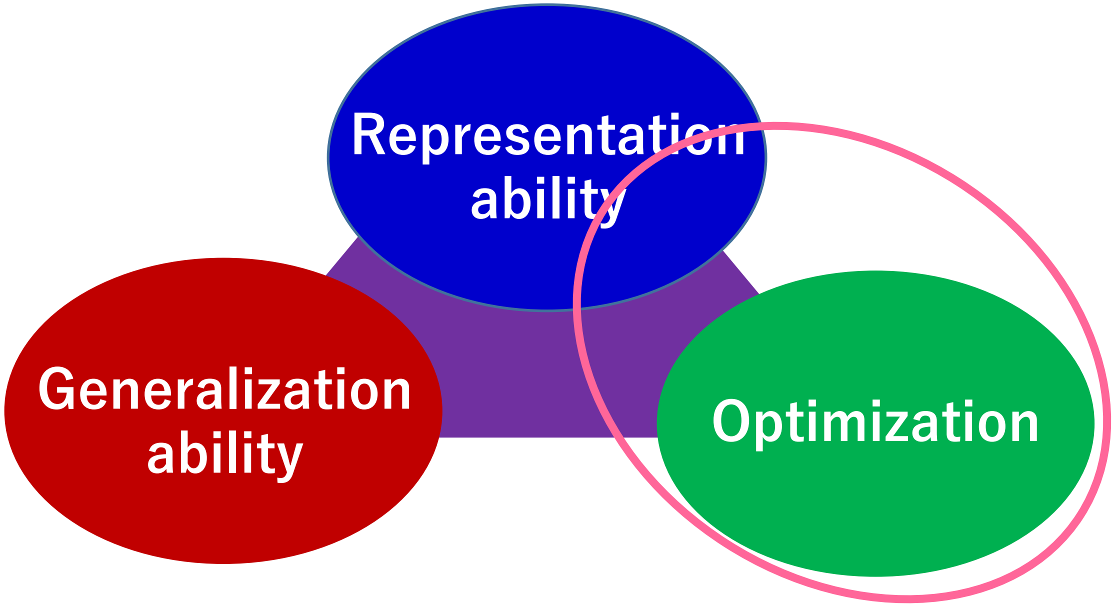
Remark : Even if the rate is better, the method does not necessarily achieve better prediction.



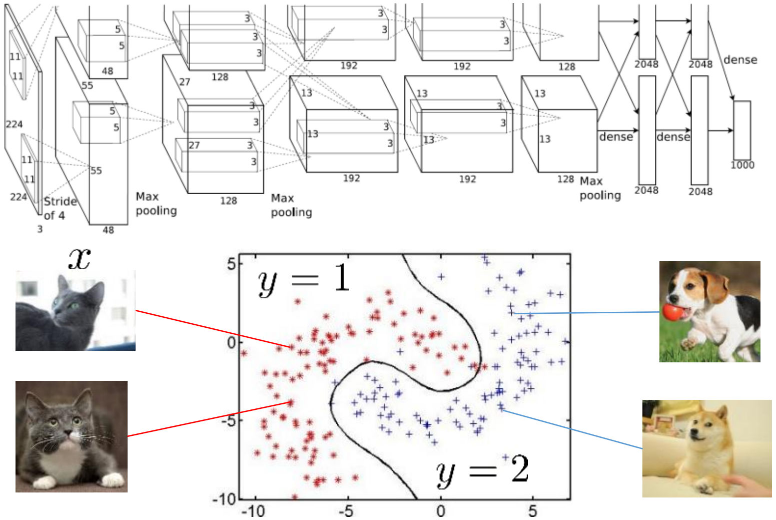
Outline

1. **Representation ability + Generalization ability**
   - Universal approximator
   - Depth separation
   - Adaptivity of deep learning
     - Inhomogeneity of smoothness
     - Curse of dimensionality
   - Foundation models
     - Diffusion model
     - Transformer
2. **Optimization ability**
   - Noisy gradient descent
   - Mean field Langevin
   - CSQ lowerbound

Optimization of NN



Optimization of NN



We should “optimize” the parameters.

$$L(W) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(W)$$

W: parameter

Loss function for the i-th data point.

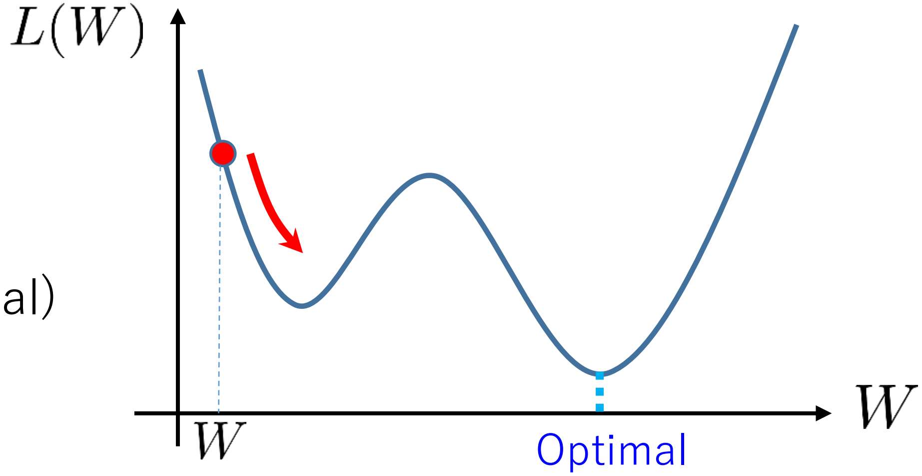
Loss function：degree of fit to the data

Loss function optimization

$$\min_{W} L(W)$$

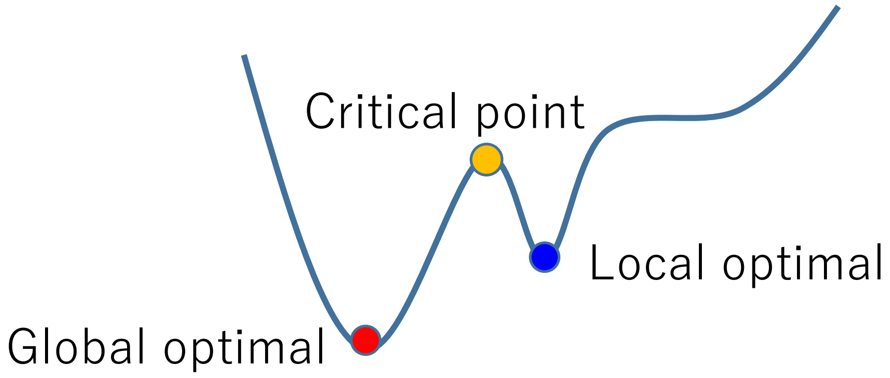
(W could be billions dimensional)

Usually, **stochastic gradient descent** is used.



Local optimality

Objective function of deep learning is <u>non-convex</u>.



- For linear deep NN, every local optimal is global optimal : Kawaguchi, 2016; Lu&Kawaguchi, 2017.

※True only for linear NN.

→ Sufficient conditions that a critical point is a global optimal was also derived by Yun, Sra&Jadbabaie (2018).

- Low rank matrix completion has no spurious local minimum : Ge, Lee&Ma, 2016; Bhojanapalli, Neyshabur&Srebro, 2016.

$$\min_{U \in \mathbb{R}^{M \times k}} \sum_{(i,j) \in E} (Y_{i,j} - (UU^\top)_{i,j})^2$$

Loss landscape

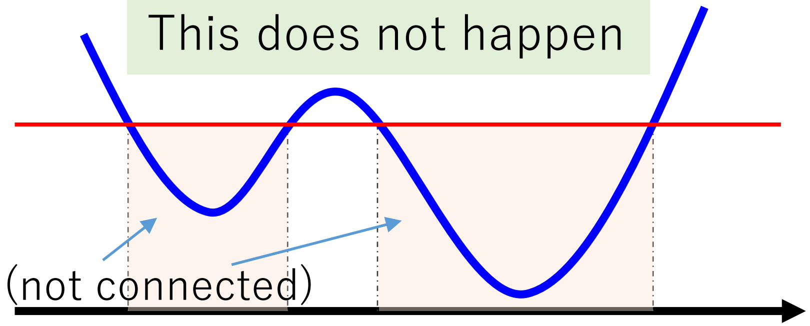
Wide neural network does not have any isolated local minimum.
(a local optimal solution is connected to global optimal solutions)

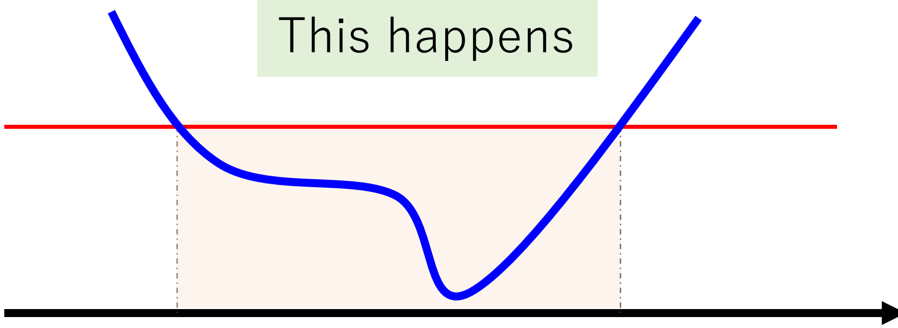
※This does not indicate GD can reach the global optimal.

Theorem

Suppose that we are given n-training data $(x_i, y_i)_{i=1}^n$, and the loss function ℓ is convex.
For two layer NN model $f_{(a,W)}(x) = \sum_{m=1}^{M} a_m \, \eta(w_m^\top x)$ with continuous activation function, if the width is not smaller than the data size $(M \geq n)$, every arcwise connected component of a level set of the empirical loss $\hat{L}(a, W) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f_{(a,W)}(x_i))$ contains the global optimal solution.

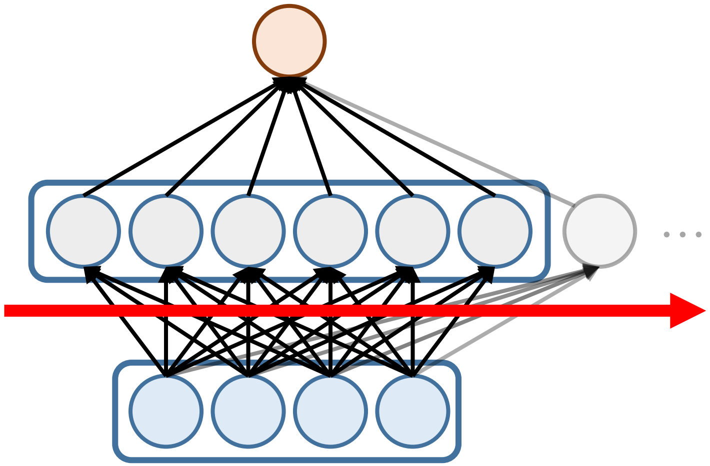
[Venturi, Bandeira, Bruna: Spurious Valleys in One-hidden-layer Neural Network Optimization Landscapes. JMLR, 20:1-34, 2019.]

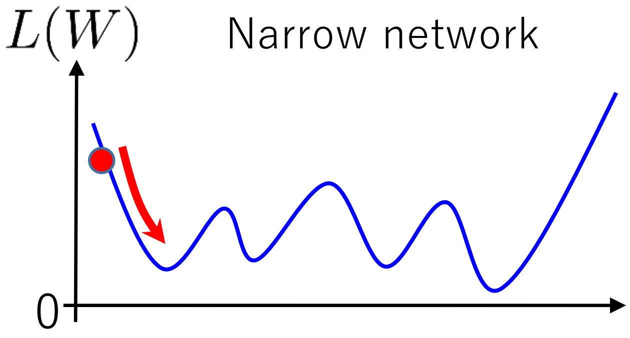


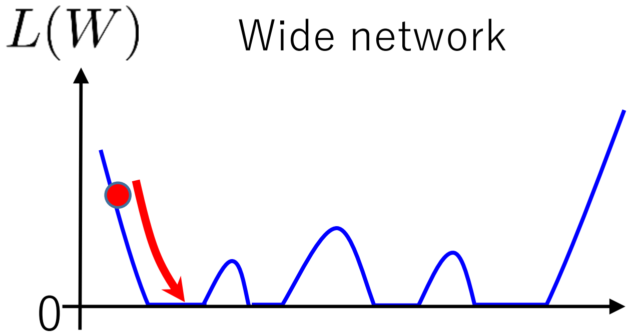


Overparameterization

Wide neural network does not have spurious local minima.







Since the model complexity is increased, the initial solution is already close to the global optimal.

- Two types of analysis
  - Neural Tangent Kernel (NTK)
  - Mean-field analysis

Two regimes

$$f_W(x) = \sum_{j=1}^{M} a_j \eta(w_j^\top x)$$

- Neural Tangent Kernel regime (lazy learning)
  - ➢ $a_j = O(1/\sqrt{M})$

[Jacot+ 2018][Du+ 2019][Arora+ 2019]
(Xavier initialization/He initialization)

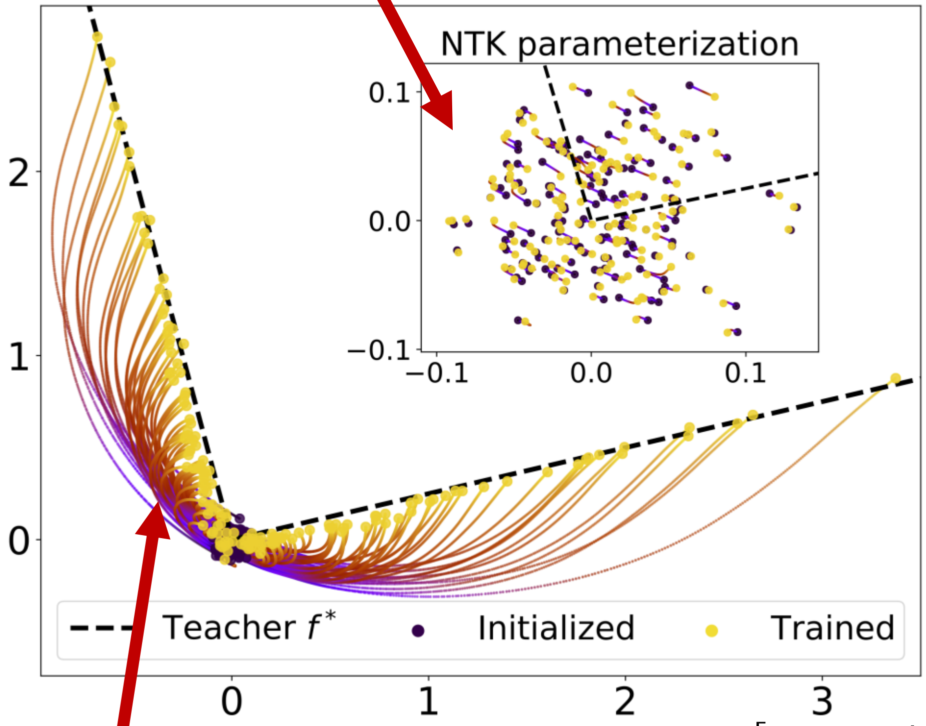
- Mean field regime
  - ➢ $a_j = O(1/M)$

[Nitanda & Suzuki (2017), Chizat & Bach (2018), Mei, Montanari, & Nguyen (2018)]

Different scaling of initial solution yields different behavior.

$$f(x) = \frac{1}{\sqrt{M}} \sum_{j=1}^{M} r_j \sigma(w_j^\top x)$$

NTK: Large scale initialization → features are (almost) freezed.



Optimization trajectory of first layer parameters in a 2-layer NN:

$$f(x) = \sum_{j=1}^{M} a_j \sigma(w_j^\top x)$$

True function:

$$f^\circ(x) = \sum_{j=1}^{2} \sigma(w_j^\top x)$$

[Ba et al., 2022]

Mean field: Small scale initialization → features need to move significantly.

$$f(x) = \frac{1}{M} \sum_{j=1}^{M} r_j \sigma(w_j^\top x)$$

ABC-parameterization

- ABC-parameterization [Yang&Hu, 2021]

$$x^l(\xi) = \phi(h^l(\xi)) \in \mathbb{R}^n, \quad h^{l+1}(\xi) = W^{l+1} x^l(\xi) \in \mathbb{R}^n, \quad \text{for } l = 1, \dots, L-1,$$

n: width

(1) Parameterization

$$W^l = n^{-a_l} w^l$$

(w^l is the actual trainable parameter)

A

(2) Initialization

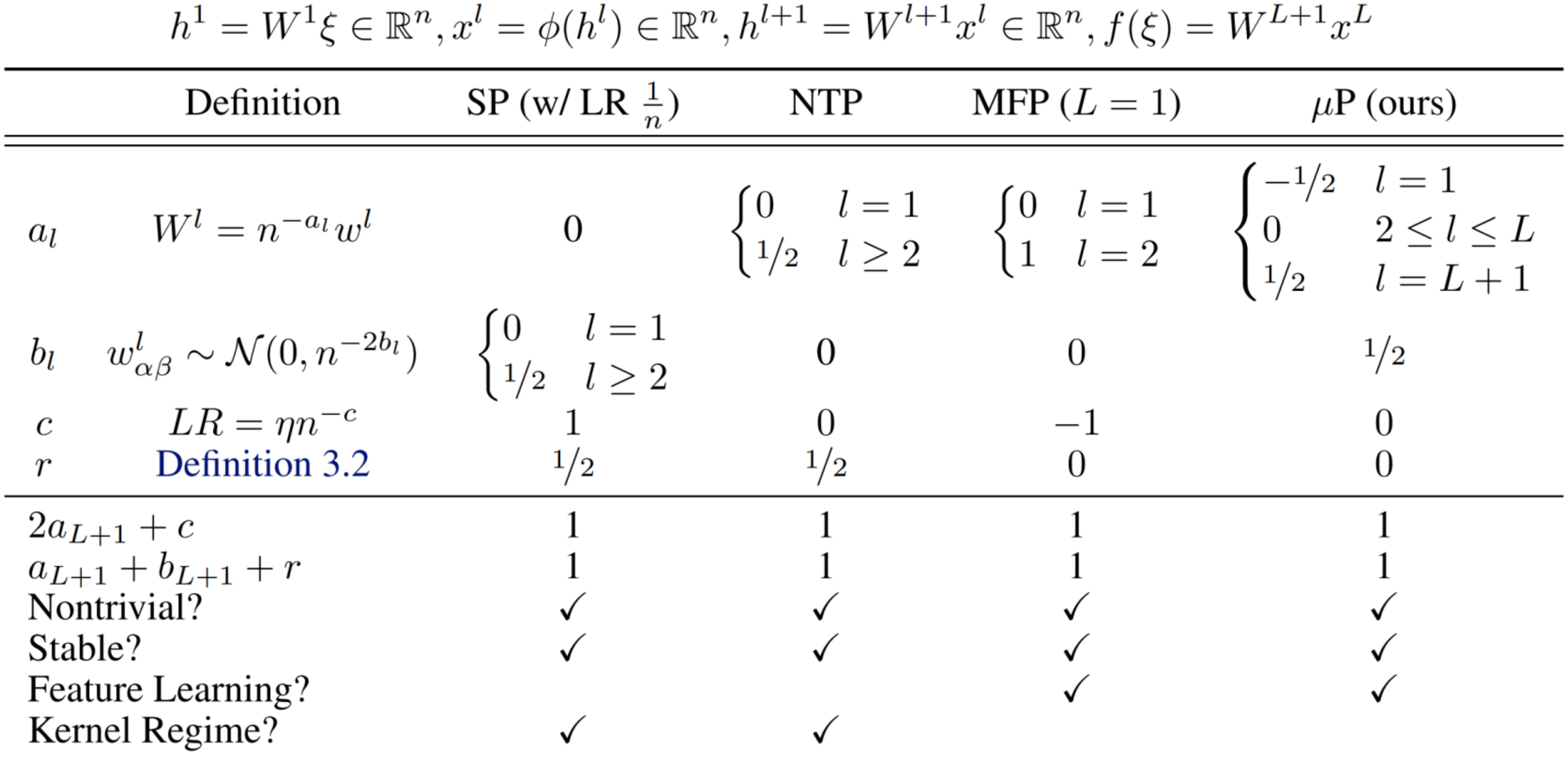
$$w^l_{\alpha\beta} \sim \mathcal{N}(0, n^{-2b_l})$$

B

(3) Learning rate

$$\eta n^{-c}$$

C



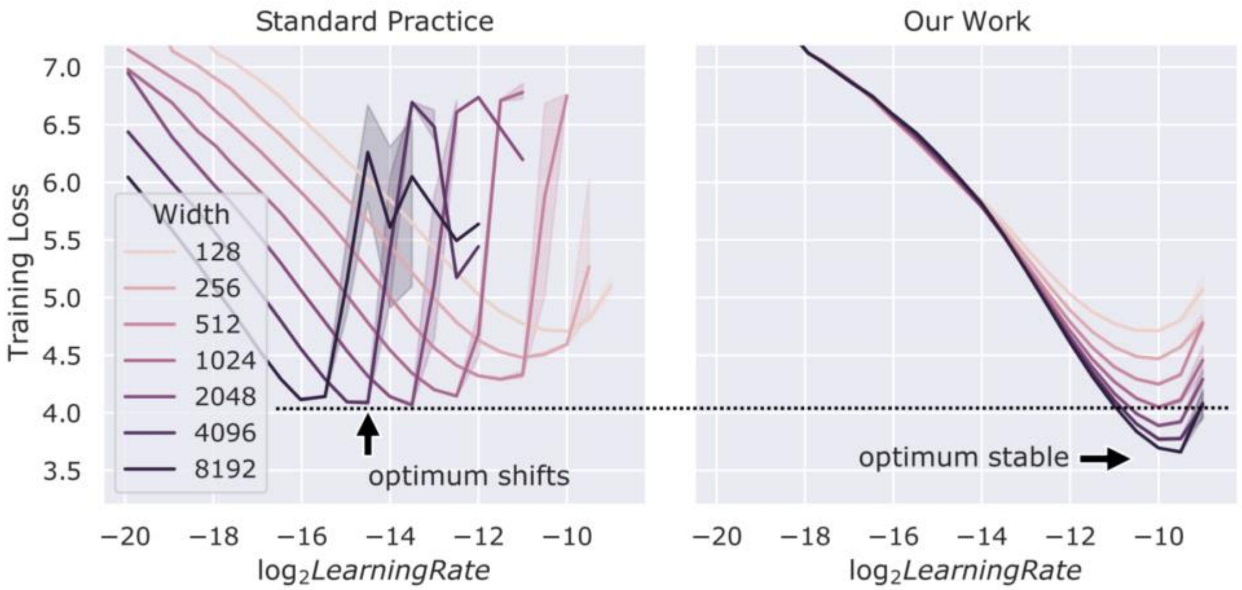
$$h^1 = W^1 \xi \in \mathbb{R}^n, x^l = \phi(h^l) \in \mathbb{R}^n, h^{l+1} = W^{l+1} x^l \in \mathbb{R}^n, f(\xi) = W^{L+1} x^L$$

	Definition	SP (w/ LR $\frac{1}{n}$)	NTP	MFP ($L=1$)	μP (ours)
a_l	$W^l = n^{-a_l} w^l$	0	$\begin{cases} 0 & l=1 \\ 1/2 & l \geq 2 \end{cases}$	$\begin{cases} 0 & l=1 \\ 1 & l=2 \end{cases}$	$\begin{cases} -1/2 & l=1 \\ 0 & 2 \leq l \leq L \\ 1/2 & l=L+1 \end{cases}$
b_l	$w^l_{\alpha\beta} \sim \mathcal{N}(0, n^{-2b_l})$	$\begin{cases} 0 & l=1 \\ 1/2 & l \geq 2 \end{cases}$	0	0	$1/2$
c	$LR = \eta n^{-c}$	1	0	-1	0
r	Definition 3.2	$1/2$	$1/2$	0	0
$2a_{L+1} + c$		1	1	1	1
$a_{L+1} + b_{L+1} + r$		1	1	1	1
Nontrivial?		✓	✓	✓	✓
Stable?		✓	✓	✓	✓
Feature Learning?				✓	✓
Kernel Regime?		✓	✓		

(Appropriate scaling)

[Yang&Hu: Tensor Programs IV: Feature Learning in Infinite-Width Neural Networks. ICML2021.]

The optimal hyper-parameter in a small size network can be transferred to huge model.
It is used to train GPT-3.5. Billions of dollars cost could be saved.



[Yang et al.:Tensor Programs V: Tuning Large Neural Networks via Zero-Shot Hyperparameter Transfer. arXiv:2203.03466]
https://github.com/microsoft/mutransformers (μP for Transformers)

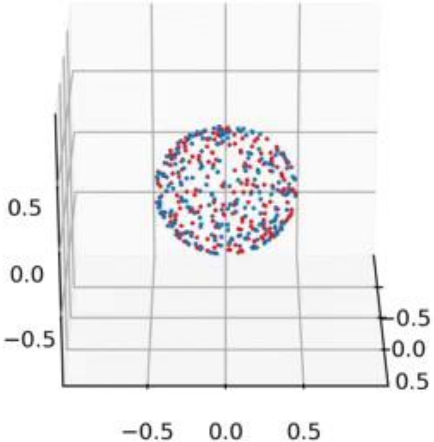
Implicit regularization

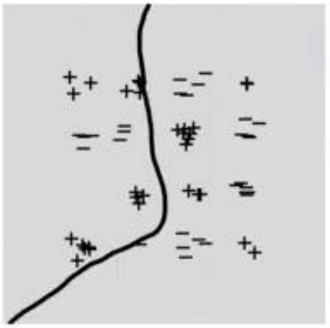
Binary classification with exp-loss:

$$\min_{\rho} \sum_{i=1}^{n} \exp\left(-y_i f_\rho(x)\right) \quad \text{where} \quad f_\rho(x) = \int \eta(w^\top x)\mathrm{d}\rho(w)$$

Optimization in the space of signed measures.

If we start from small initialization, only neurons that are necessary for classification “grow up.”





[Chizat&Bach: Implicit Bias of Gradient Descent for Wide Two-layer Neural Networks Trained with the Logistic Loss. COLT2020.]

Optimization dynamics implicitly regularize the solution.

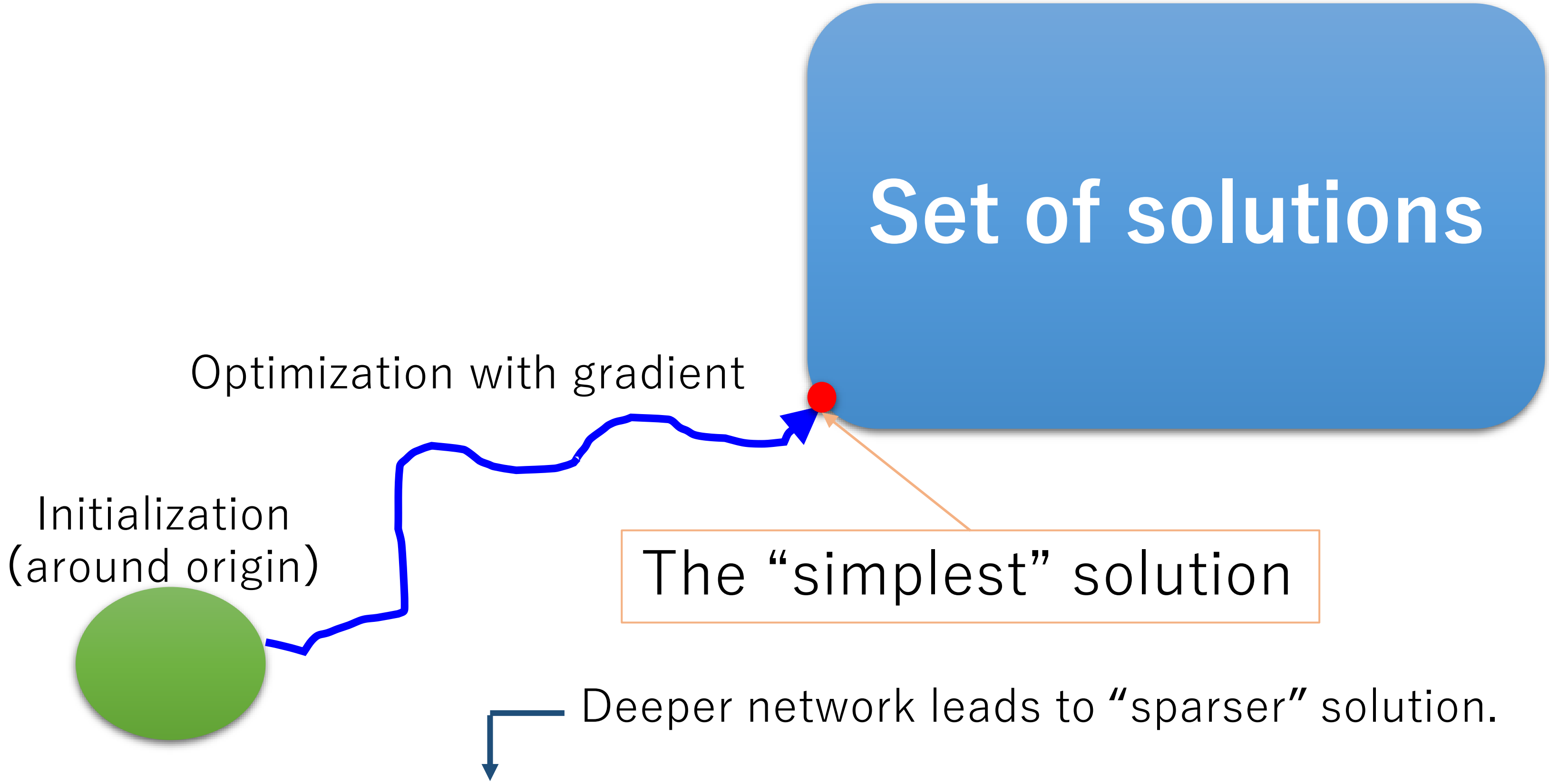
→ **Sparse solution：implicit regularization**

The solution converges to the max-margin solution under L1-norm constraint (if the sequence converges to a “global minimizer” direction):

$$\max_{\rho : \|\rho\|_{\mathcal{F}_1} \leq 1} \min_{i \in \{1,\dots,n\}} y_i f_\rho(x_i) \qquad \|\rho\|_{\mathcal{F}_1} = |\rho|(\mathbb{R}^d)$$

Gradient descent and implicit regularization

- Dynamics starting from a small initialization converges to the minimum norm solution.
  → Implicit regularization



Deeper network leads to "sparser" solution.

[Gunasekar et al.: Implicit Regularization in Matrix Factorization, NIPS2017]
[Soudry et al.: The implicit bias of gradient descent on separable data. JMLR2018]
[Gunasekar et al.: Implicit Bias of Gradient Descent on Linear Convolutional Networks, NIPS2018]
[Moroshko et al.: Implicit Bias in Deep Linear Classification: Initialization Scale vs Training Accuracy, arXiv:2007.06738]

Implicit regularization in each regime

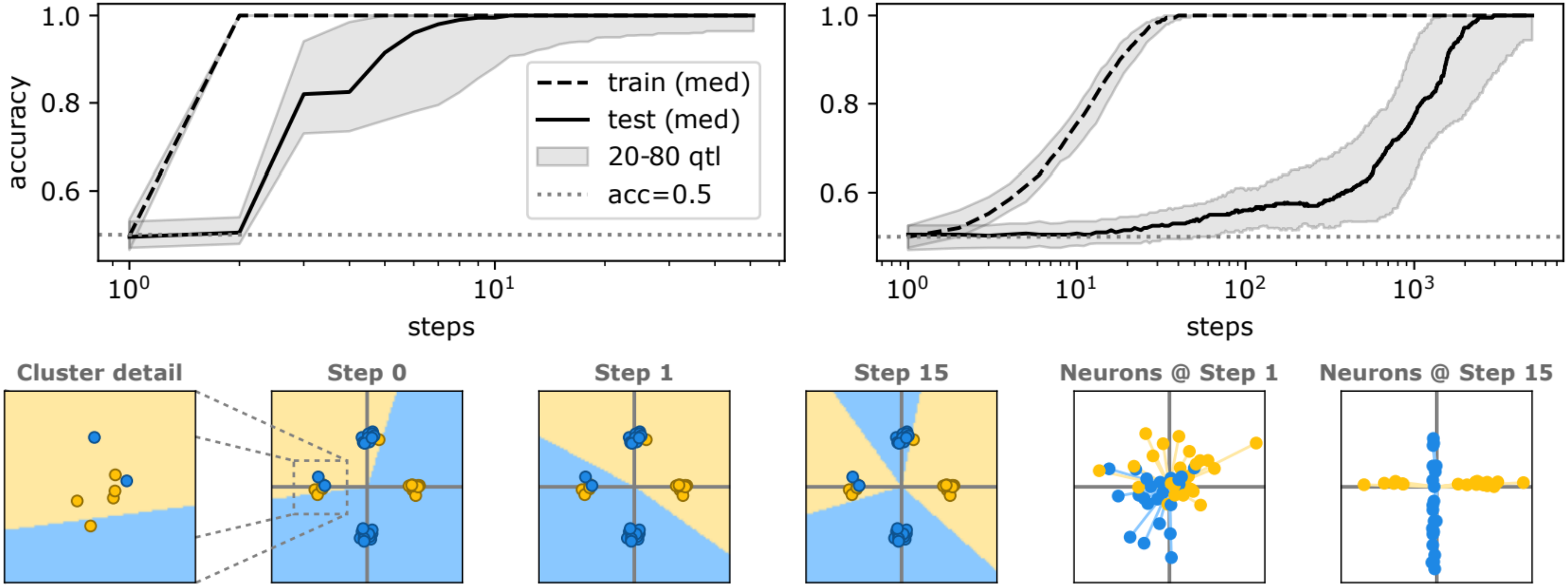
Regime	Implicit regularization
NTK, kernel method with early stopping	L2-regularization
Mean-field	L1-regularization

- Deep learning uses several **“explicit regularization”**.
  → Batch normalization, Dropout, Weight decay, MixUp, …
- On the other hand, the **“implicit regularization”** induced by the deep structure and optimization dynamics is also very important.
  → Benign overfitting, Grokking, Flat-minimum, …

Grokking / Benign-overfitting

Grokking: [Power et al.: Grokking: Generalization beyond overfitting on small algorithmic datasets. arXiv:2201.02177]

[Xu et al.: Benign Overfitting and Grokking in ReLU Networks for XOR Cluster Data. arXiv2310.02541]



It is also called “hidden progress” [Barak et al. 2022].

See also [Meng et al.: Benign Overfitting in Two-Layer ReLU Convolutional Neural Networks for XOR Data. arXiv2310.01975]

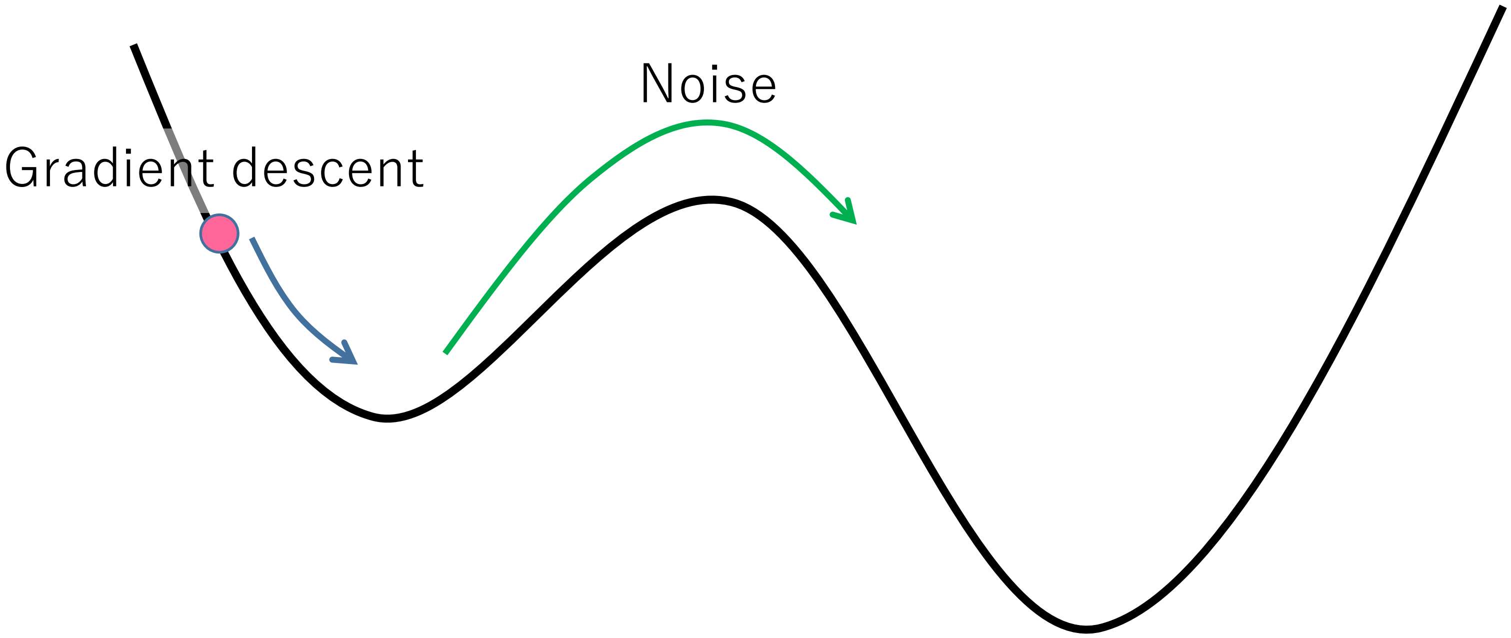
Outline

1. **Representation ability**
   - Universal approximator
   - Adaptivity of deep learning
     - Inhomogeneity of smoothness
     - Curse of dimensionality
2. **Generalization ability**
   - Double descent, Benign overfitting for overparameterized model
   - Generalization gap analysis
     - Norm based bound
     - Compression based bound
3. **Optimization ability**
   - Neural Tangent Kernel
   - Dynamics in a feature learning regime
   - Mean field analysis

Noisy gradient descent and its global optimality

Noisy gradient descent

The model is not linearly approximated.
We need to solve “non-convex” optimization.

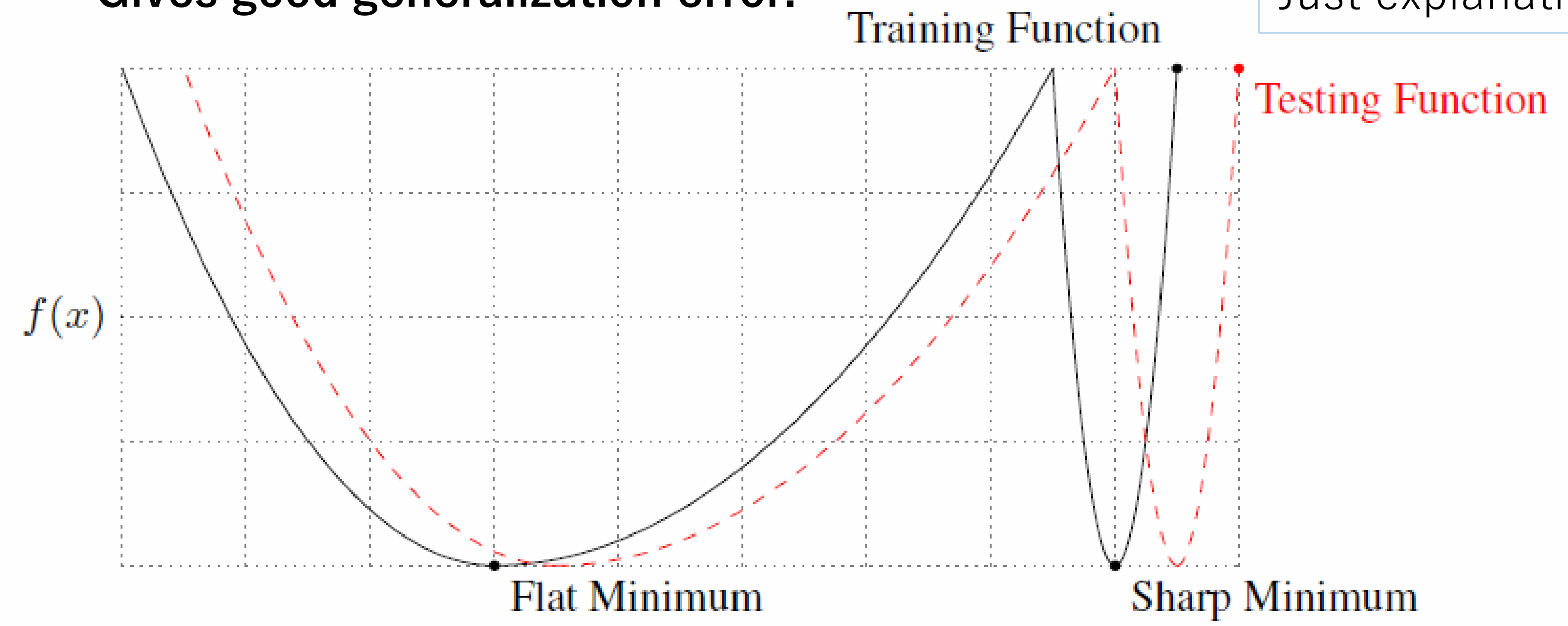


SGD is a noisy gradient descent.
Noisy perturbation is helpful to escape local minimum.

Sharp minima vs flat minima

It is said that SGD likely stay in “flat local minimum”
→ Gives good generalization error.

This is not theory,
Just explanation.



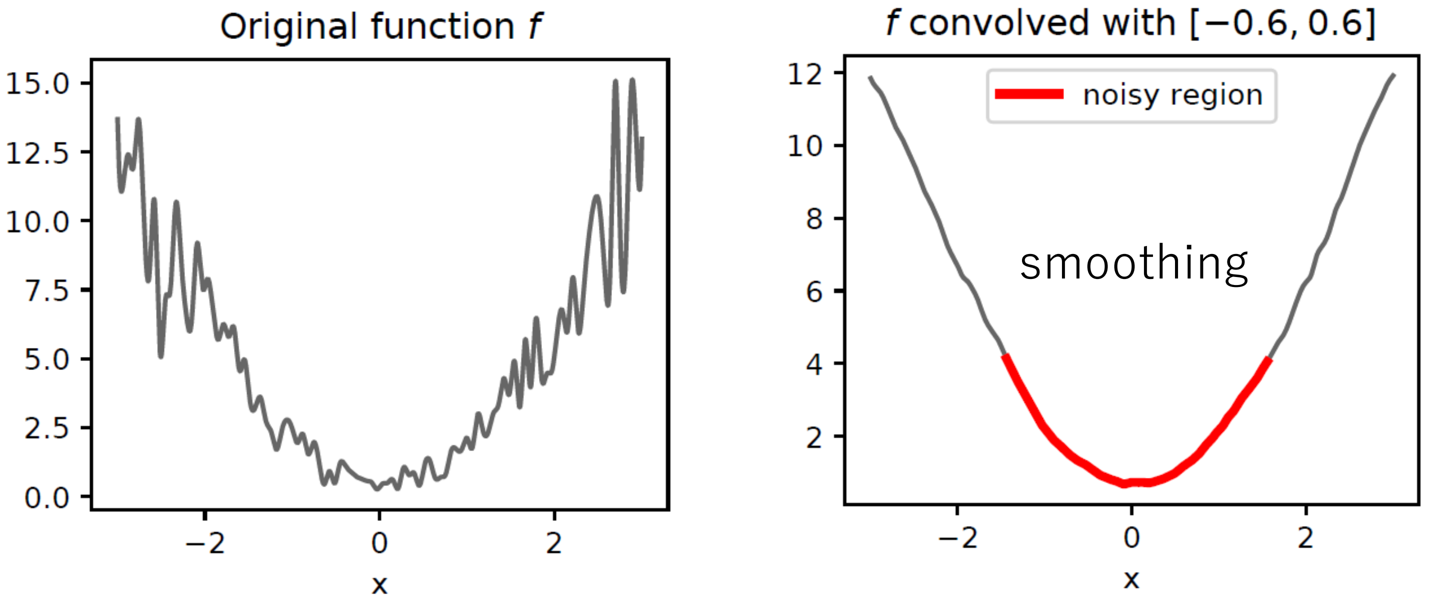
Keskar, Mudigere, Nocedal, Smelyanskiy, Tang (2017):
On large-batch training for deep learning: generalization gap and sharp minima.

$$\theta_t = \theta_{t-1} - \alpha_b \underbrace{\left(\frac{1}{b} \sum_{j=1}^{b} \nabla_\theta \ell(z_{i_j};\theta) \right)}_{\cong \text{Normal distribution}}$$

→ Random walk is likely to be captured in a flat region.

- (criticism) The concept of “flat” depends on the choice of coordinate system. (Dinh et al., 2017)
- PAC-Bayesian analysis (Dziugaite, Roy, 2017)

Smoothing by noisy gradient



[Kleinberg, Li, and Yuan, ICML2018]

Stochastic gradient ⇒ Noise is added ⇒ Objective is smoothed

$$x_t = x_{t-1} - \eta(\nabla L(x_{t-1}) + \xi_t) \qquad (y_t = x_t + \eta\xi_t)$$

$$\Rightarrow y_t = y_{t-1} - \eta\xi_{t-1} - \eta\nabla L(y_{t-1} - \eta\xi_{t-1})$$

$$\Rightarrow \mathbb{E}_{\xi_{t-1}}[y_t] = y_{t-1} - \eta\nabla\mathbb{E}_{\xi_{t-1}}[L(y_{t-1} - \eta\xi_{t-1})]$$

SGD optimizes a "smoothed" objective: $\bar{L}(y_t) = \mathbb{E}_{\xi_t}[L(y_t - \eta\xi_t)]$

GLD/SGLD

- Stochastic Gradient Langevin Dynamics (SGLD)

$$\min_{x\in\mathbb{R}^d} L(x) = \min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \ell_i(x)$$

(Non-convex)

$$\mathrm{d}X_t = -\nabla L(X_t)\mathrm{d}t + \sqrt{2\lambda}\mathrm{d}B_t$$

(Gradient Langevin dynamics)

Stationary distribution : $\pi \propto \exp(-L(X)/\lambda)$

Discretization

[Gelfand and Mitter (1991); Borkar and Mitter (1999); Welling and Teh (2011)]

GLD: $X_{t+1} = X_t - \eta\nabla L(X_t) + \sqrt{2\eta\lambda}\xi_t$ (Euler-Maruyama scheme)

$\xi_t \sim \mathrm{N}(0, I)$

SGLD: $X_{t+1} = X_t - \eta\underbrace{\frac{1}{|I_B|}\sum_{i\in I_B}\nabla\ell_i(X_t)}_{\text{Stochastic gradient}} + \sqrt{2\eta\lambda}\xi_t$

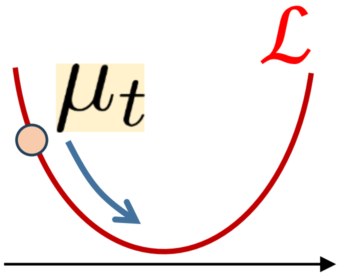
GLD as a Wasserstein gradient flow

$$\mathrm{d}X_t = -\nabla L(X_t)\mathrm{d}t + \sqrt{2\lambda}\mathrm{d}B_t$$

μ_t : Distribution of X_t (we can assume it has a density)

PDE that describes μ_t's dynamics [**Fokker-Planck equation**]:

$$\begin{aligned}\partial_t \mu_t &= \nabla \cdot [\mu_t \nabla L] + \lambda \Delta_x \mu_t \\ &= \nabla \cdot [\mu_t (\nabla L + \lambda \nabla \log(\mu_t))]\end{aligned}$$



This is the **Wasserstein gradient flow** to minimize the following objective:

$$\mu^* = \underset{\mu \in \mathcal{P}}{\arg\min} \underbrace{\int L(x)\mathrm{d}\mu(x)}_{\text{[linear w.r.t. } \mu \text{]}} + \lambda \mathrm{Ent}(\mu) =: \mathcal{L}(\mu)$$

$(\mathrm{Ent}(\mu) = \int \log(\mu)\mathrm{d}\mu)$

➡ $\mu_t \rightsquigarrow \mu^*(x) \propto \exp(-L(x)/\lambda)$ = Stationary distribution

c.f., Donsker-Varadan duality formula

Continuity equation

Continuity equation: $$\frac{\partial \mu_t}{\partial t} = -\nabla \cdot (v_t \mu_t)$$ $v_t(x) = -(\nabla L(x) + \lambda \nabla \log(\mu_t)(x))$

The meaning of this equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\int f(x)\mu_t(x)\mathrm{d}x = \int (\nabla f(x))^\top v_t(x)\mu_t(x)\mathrm{d}x$$

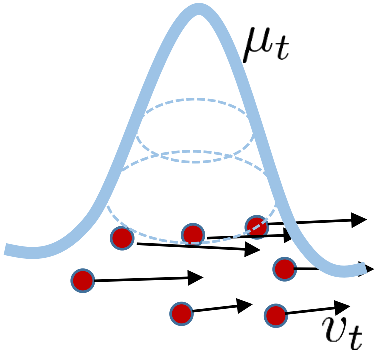
$$\left(= -\int f(x)\nabla \cdot (v_t \rho_t)\mathrm{d}x\right)$$

($\forall f$: compact support, C^∞-class)

- Let T_t be a map generated by the vector field v_t: $\frac{\mathrm{d}T_t}{\mathrm{d}t}(x) = v_t(T_t(x))$.
- μ_t is the push-forward of μ_0 by a map $T_t: \mathrm{R}^d \to \mathrm{R}^d$: $\mu_t = T_{t\#}\mu_0$.
  That is, $\boldsymbol{\mu_t}$ is the distribution of $\boldsymbol{T_t(x)}$ where $\boldsymbol{x \sim \mu_0}$.

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int f(x)\mu_t(x)\mathrm{d}x &= \frac{\mathrm{d}}{\mathrm{d}t}\int f(T_t(x))\mu_0(x)\mathrm{d}x \\ &= \int \nabla f(T_t(x))^\top \frac{\mathrm{d}T_t(x)}{\mathrm{d}t}\mu_0(x)\mathrm{d}x \\ &= \int \nabla f(T_t(x))^\top v_t(T_t(x))\mu_0(x)\mathrm{d}x \\ &= \int \nabla f(x)^\top v_t(x)\mu_t(x)\mathrm{d}x.\end{aligned}$$

[continuity equation]

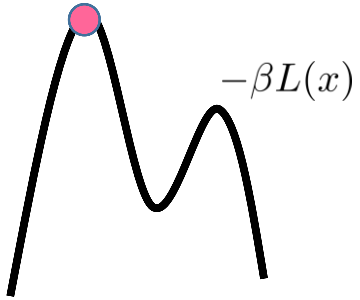


Stationary distribution

$$\mathrm{d}X_t = -\nabla L(X_t)\mathrm{d}t + \sqrt{2\beta^{-1}}\mathrm{d}B_t$$

Stationary distribution of the continuous time dynamics:

$$\mu^*(\mathrm{d}x) \propto \exp(-\beta L(x))\mathrm{d}x$$



The stationary distribution concentrates around the optimal solution.

Wasserstein gradient flow

$$\boxed{\mu^*(x) \propto \exp(-\lambda^{-1}L(x))}$$

$$\begin{aligned}
\lambda^{-1}\mathcal{L}(\mu) &= \int \lambda^{-1}L(x)\mathrm{d}\mu(x) + \mathrm{Ent}(\mu) \\
&= \int -\log(\mu^*)\mathrm{d}\mu + \int \log(\mu)\mathrm{d}\mu + (\mathrm{const.}) \\
&= \int \log\left(\frac{\mu}{\mu^*}\right)\mathrm{d}\mu = \mathrm{KL}(\mu||\mu^*)
\end{aligned}$$

We neglect this term below

By the continuity equation $\mu_t = -\nabla \cdot [v_t\mu_t]$, it holds that

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{KL}(\mu_t||\mu^*) &= \frac{\mathrm{d}}{\mathrm{d}t}\int \log\left(\frac{\mu_t(x)}{\mu^*(x)}\right)\mu_t(x)\mathrm{d}x \\
&= \int \log\left(\frac{\mu_t(x)}{\mu^*(x)}\right)\partial_t\mu_t(x)\mathrm{d}x + \underbrace{\int \frac{\partial_t\mu_t(x)}{\cancel{\mu_t(x)}}\cancel{\mu_t(x)}\mathrm{d}x}_{=0} \\
&= \int \log\left(\frac{\mu_t(x)}{\mu^*(x)}\right)\nabla\cdot(-v_t\mu_t(x))\mathrm{d}x \\
&= \int \langle v_t, \nabla\log(\mu_t) - \nabla\log(\mu^*)\rangle\mu_t(x)\mathrm{d}x
\end{aligned}$$

Wasserstein勾配流

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{KL}(\mu_t||\mu^*) = \int \langle v_t, \nabla \log(\mu_t) - \nabla \log(\mu^*)\rangle \mu_t(x)\mathrm{d}x$$

In particular, if

$$v_t = -\left(\lambda\nabla \log(\mu_t) + \nabla L\right) = -\lambda\left(\nabla \log(\mu_t) - \nabla \log(\mu^*)\right) \quad \text{(GLD)}$$

then this is the steepest gradient descent direction such that

$$\partial_t \mu_t = \nabla \cdot \left[\underbrace{(\lambda \nabla \log(\mu_t) + \nabla L)}_{=: -v_t} \mu_t\right]$$

$$\boxed{\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{KL}(\mu_t||\mu^*) &= -\lambda \int \|\nabla \log(\mu^*) - \nabla \log(\mu_t)\|^2 \mu_t \mathrm{d}x \\ &= -\lambda I(\mu_t||\mu^*)\end{aligned}}$$

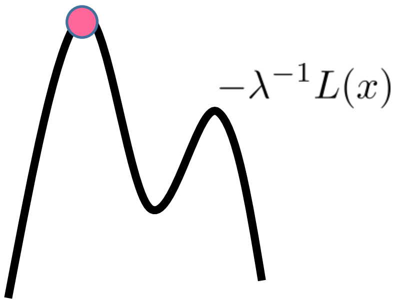
Fisher divergence:

$$I(\mu||\nu) := \int \|\nabla \log(\nu) - \nabla \log(\mu)\|^2 \mu(x)\mathrm{d}x$$

GLD is the Wasserstein gradient flow to minimize the KL-div from μ^*.

Log-Sobolev inequality

Stationary distribution: $\mu^*(x) \propto \exp(-\lambda^{-1}L(x))$



Def (log-Sobolev inequality)

There exists a constant $\alpha > 0$ such that
for any probability measure ν (absolutely-continuous w.r.t. μ^*)

$$\mathrm{KL}(\nu||\mu^*) \leq \frac{1}{2\alpha} I(\nu||\mu^*)$$

E.g. :

- Quadratic+Bounded
- Weak Morse function

KL-div

Fisher-div

$$\mathrm{KL}(\nu||\mu) = \int \log\left(\frac{\nu}{\mu}\right)\mu\mathrm{d}x, \quad I(\nu||\mu) = \int \left\|\nabla \log \frac{\nu}{\mu}\right\|^2 \nu\mathrm{d}x$$

Geometric ergodicity μ_t: distribution of X_t

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{KL}(\mu_t||\mu^*) = -\lambda I(\mu_t||\mu^*) \leq -2\alpha \mathrm{KL}(\mu_t||\mu^*)$$

(by log-Sobolev)

$$\mathrm{KL}(\mu_t||\mu^*) \leq \exp(-2\alpha t)\mathrm{KL}(\mu_0||\mu^*)$$

Linear convergence w.r.t. KL-div

[Bakry, Gentil, and Ledoux: *Analysis and Geometry of Markov Diffusion Operators*. Springer, 2014. Th. 5.2.1]

Sufficient condition for log-Sobolev inequality

Strongly convex (Bakry-Emery criterion):

$\mu^*(x) \propto \exp(-\lambda^{-1}L(x))$

$$\nabla\nabla^\top L(x) \succcurlyeq \mu I \quad \Rightarrow \quad \alpha \geq \mu/\lambda$$

[Bakry and Émery, 1985]

Ex.: OU-process. $L(x) = \frac{x^2}{2} \Rightarrow \mu = 1$

Bounded perturbation lemma (Hollley-Stroock):

Suppose that $\mu^*(x) = \mu(x)\exp(h(x))$ and μ satisfies α'-LSI, then

$|h(x)| \leq B\ (\forall x)$ ➡ μ^* satisfies α-LSI with $\alpha \geq \alpha' \exp(-4B)$

[R. Holley and D. Stroock. Logarithmic sobolev inequalities and stochastic Ising models. Journal of statistical physics, 46(5-6):1159–1194, 1987.]

Ex.: $L(x) = \ell(x) + \lambda_1 x^2$ and $|\ell(x)| \leq B$, then μ^* satisfies LSI with $\alpha = \frac{2\lambda_1}{\lambda}\exp(-4B/\lambda)$.

$\mu^*(x) \propto \exp(-\lambda^{-1}L(x))$

References

- **Finite dimensional Langevin dynamics:**

➢ Convergence in low (convex case): Dalalyan and Tsybakov, 2012; Dalalyan, 2016; Durmus and Moulines, 2015, ..

➢ Non-convex Optimization: Raginsky et al., 2017; Xu et al., 2018; Erdogdu, Mackey and Shamir, 2018

➢ Log-Sobolev inequality: Vempala and Wibisono, 2019.

- **Infinite dimensional Langevin dynamics:**

➢ Continuous time:

  - Existence & Uniqueness of invariant measure: Da Prato and Zabczyk,1992; Maslowski, 1989; Sowers, 1992.
  - Geometric ergodicity: Jacquot and Royer, 1995; Shardlow, 1999; Hairer, 2002, Its explicit rate: Goldys and Maslowski, 2006.

➢ Discrete time:

  - Weak approximation rate of discretized scheme: Hausenblas, 2003; Debussche, 2011; Bréhier, 2014; Bréhier and Kopec 2016.

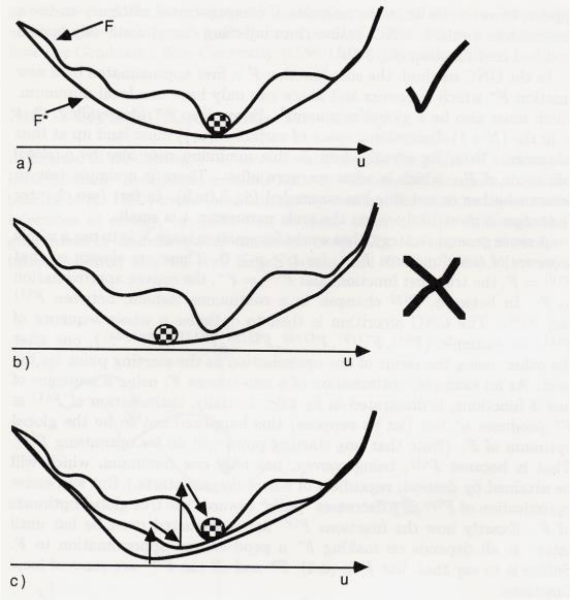
Other topics (MCMC in Hilbert space):

  - preconditioned Crank–Nicolson (pCN): Hairer et al., 2014; Eberle, 2014; Vollmer, 2015; Rudolf and Sprungk, 2018.
  - Metropolis-Adjusted Langevin Algorithm (MALA): Durmus and Moulines, 2015; Beskos et al., 2017.

Related work: Graduated optimization

- Graduated non-convexity

Blake and Zisserman: *Visual reconstruction*, volume 2. MIT press Cambridge, 1987.



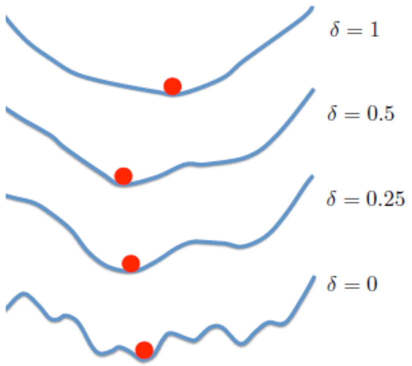
- Convolution with Gaussian kernel

Z. Wu. The effective energy transformation scheme as a special continuation approach to global optimization with application to molecular conformation. SIAM Journal on Optimization, 6(3):748-768, 1996.

- Graduated optimization

Hazan, Levy, and Shalev-Shwartz: On graduated optimization for stochastic non-convex problems. *International conference on machine learning*, pp. 1833-1841, 2016.

  - σ-nice property: $\hat{L}_\delta(x) = \mathrm{E}_{u\sim U(\mathrm{B}(\mathrm{R}^d))}[L(x+\delta u)]$
  - Polynomial time convergence.

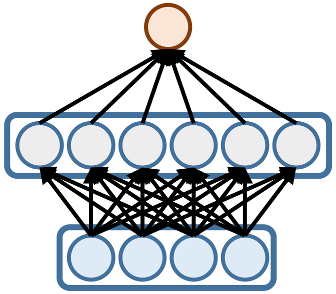


Survey:
Mobahi and Fisher III. On the link between gaussian homotopy continuation and convex envelopes. *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 43-56, 2015.

Optimization theory
in mean field regime

2-layer NN in mean-field scaling

- 2-layer neural network:



$$f(z) = \frac{1}{M}\sum_{j=1}^{M} r_j \sigma(w_j^\top z)$$

Non-linear with respect to parameters $(r_j, w_j)_{j=1}^{M}$.

$$f_{\mathscr{X}}(z) = \frac{1}{M}\sum_{j=1}^{M} h_{X^{(j)}}(z)$$

where $X^{(j)} = (r_j, w_j)$ and $h_x(z) = r\sigma(w^\top z)$ for $x = (r, w)$.

Loss function:

$$F(\mathscr{X}) = \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell_i(f_{\mathscr{X}}(z_i))}_{\text{loss}} + \lambda_1 \underbrace{\frac{1}{M}\sum_{j=1}^{M} \|X^{(j)}\|^2}_{\text{L2 regularization}}$$

Non-convex

Application of GLD

$$F(\mathscr{X}) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(f_{\mathscr{X}}(z_i)) + \lambda_1 \frac{1}{M}\sum_{j=1}^{M} \|X^{(j)}\|^2$$

$$f_{\mathscr{X}}(z) = \frac{1}{M}\sum_{j=1}^{M} h_{X^{(j)}}(z)$$

Noisy gradient descent update:

$$X_{k+1}^{(j)} = X_k^{(j)} - \eta_k \nabla_{X_k^{(j)}} F(\mathscr{X}_k) + \sqrt{2\eta_k\lambda_2}\xi_k^{(j)} \qquad \xi_k^{(j)} \sim \mathrm{N}(0, I)$$

$$\Leftrightarrow \quad X_{k+1}^{(j)} = X_k^{(j)} - \frac{\eta_k}{M}\left(\frac{1}{n}\sum_{i=1}^{n} \ell_i'(f_{\mathscr{X}_k}(z_i))\nabla_{X^{(j)}} h_{X_k^{(j)}}(z_i) + \lambda_1 X_k^{(j)}\right) + \sqrt{2\eta_k\lambda_2}\xi_k^{(j)}$$

Does it converge?

Naïve application of existing theory in gradient Langevin dynamics yields

$$K = \exp\left(\mathrm{O}(Md)\right)\log(1/\epsilon)$$

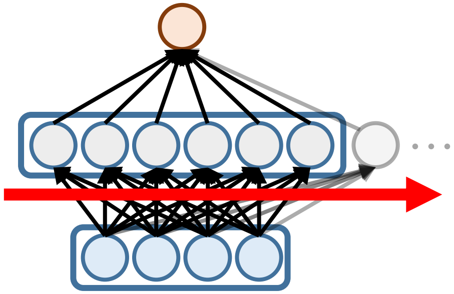
iteration complexity to achieve ϵ error.
→ Cannot be applied to wide neural network.

[Raginsky, Rakhlin and Telgarsky, 2017; Xu, Chen, Zou, and Gu, 2018; Erdogdu, Mackey and Shamir, 2018; Vempala and Wibisono

Mean field limit

$$f(z) = \frac{1}{M}\sum_{j=1}^{M} r_j\sigma(w_j^\top z)$$

Non-linear with respect to the parameters $(r_j, w_j)_{j=1}^{M}$.



★Mean field limit:

$$f(z) = \frac{1}{M}\sum_{j=1}^{M} r_j\sigma(w_j^\top z) \xrightarrow{M\to\infty} \boxed{f_\mu(z) = \int r\sigma(w^\top z)\mathrm{d}\mu(r,w)}$$

Linear with respect to μ.

[Nitanda&Suzuki, 2017][Chizat&Bach, 2018][Mei, Montanari&Nguyen, 2018][Rotskoff&Vanden-Eijnden, 2018]

Loss function (empirical risk + regularization):

$$F(\mathscr{X}) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(f_{\mathscr{X}}(z_i)) + \lambda_1\frac{1}{M}\sum_{j=1}^{M}\|X^{(j)}\|^2$$

➡ $$F(\mu) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(f_\mu(z_i)) + \lambda_1\mathbb{E}_\mu[\|x\|^2]$$

Convex w.r.t. μ if the loss ℓ_i is convex (e.g., squared / logistic loss).

General form of mean field LD

$$\mathcal{L}(\mu) := \underline{F(\mu)} + \lambda_2 \mathrm{Ent}(\mu)$$

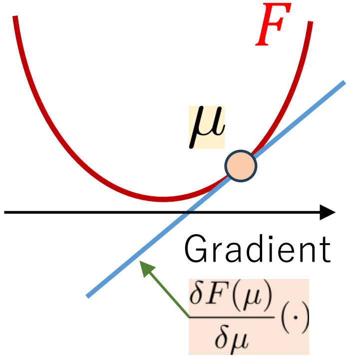
convex + strictly convex = strictly convex

$F(\theta\mu + (1-\theta)\nu) \leq \theta F(\mu) + (1-\theta)F(\nu)$ $\quad$ $(\mathrm{Ent}(\mu) = \int \log(\mu)\mathrm{d}\mu)$

Mean field Langevin dynamics:

- SDE the Fokker-Planck equation of which corresponds to the Wasserstein GF:

$$\mathrm{d}X_t = -\nabla \frac{\delta F(\mu_t)}{\delta \mu}(X_t)\mathrm{d}t + \sqrt{2\lambda_2}\mathrm{d}B_t$$
$$\mu_t = \mathrm{Law}(X_t)$$



GLD: $\mathrm{d}X_t = -\nabla L(X_t)\mathrm{d}t + \sqrt{2\lambda_2}\mathrm{d}B_t$, $\quad \frac{\delta F(\mu)}{\delta \mu}(\cdot) = L(\cdot)$

$F(\mu) = \int L(x)\mathrm{d}\mu$

Definition (first variation)

The first variation $\frac{\delta F}{\delta \mu}: \mathcal{P} \times \mathbb{R}^d \to \mathbb{R}$ is defined as a continuous functional such as

$$\lim_{\epsilon \to 0} \frac{F(\epsilon\nu + (1-\epsilon)\mu) - F(\mu)}{\epsilon} = \int \frac{\delta F(\mu)}{\delta \mu}(x)\mathrm{d}(\nu - \mu)(x)$$

MF-LD to optimize mean field NN

Loss function: $F(\mu) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(f_\mu(z_i)) + \lambda_1 \mathbb{E}_\mu[\|X\|^2]$

$$\boxed{f_\mu(z) = \int h_x(z)\mathrm{d}\mu(x)}$$

$$X^{(j)}_{k+1} = X^{(j)}_k - \frac{\eta_k}{M}\left(\frac{1}{n}\sum_{i=1}^{n} \ell_i'(\underline{f_{\mathscr{X}_k}}(z_i))\nabla_{X^{(j)}} h_{X^{(j)}_k}(z_i) + \lambda_1 X^{(j)}_k\right) + \sqrt{2\eta_k\lambda_2}\,\xi^{(j)}_k$$

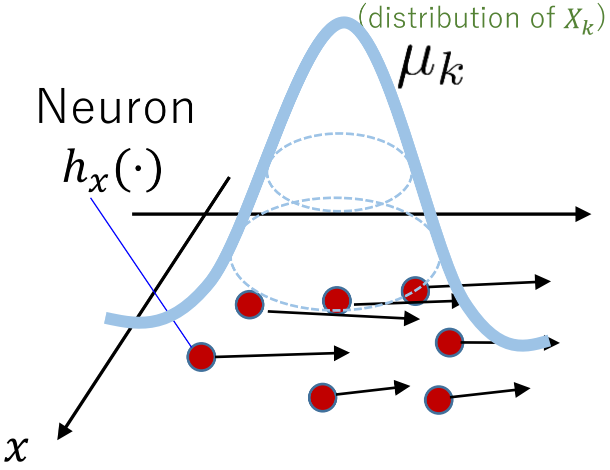
$$\Rightarrow \quad X_{k+1} = X_k - \eta_k \underbrace{\left(\frac{1}{n}\sum_{i=1}^{n} \ell_i'(\underline{f_{\mu_k}(z_i)})\nabla_X h_{X_k}(z_i) + \lambda_1 X_k\right)}_{\nabla \frac{\delta F(\mu_k)}{\delta \mu}(X_k)} + \sqrt{2\eta_k\lambda_2}\,\xi_k$$

$$\mu_k = \mathrm{Law}(X_k)$$

$$\boxed{\frac{\delta F(\mu)}{\delta \mu}(X) = \frac{1}{n}\sum_{i=1}^{n} \ell_i'(f_\mu(z_i))h_X(z_i) + \lambda_1\|X\|^2}$$

Discrete time MFLD:

$$X_{k+1} = X_k - \eta_k \nabla \frac{\delta F(\mu_k)}{\delta \mu}(X_k) + \sqrt{2\eta_k\lambda_2}\,\xi_k$$



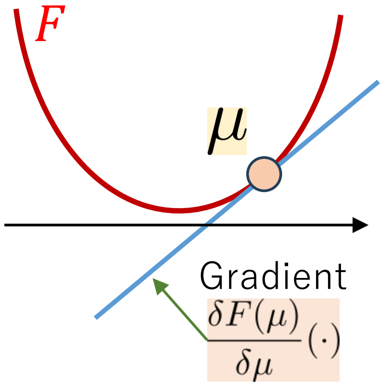
Proximal Gibbs measure

$$\mathcal{L}(\mu) = \underline{F(\mu)} + \lambda_2 \mathrm{Ent}(\mu)$$

$$(\mathrm{Ent}(\mu) = \textstyle\int \log(\mu)\mathrm{d}\mu)$$

Linearized objective at μ:

$$\bar{\mathcal{L}}_\mu(\nu) = \underline{\int \frac{\delta F(\mu)}{\delta \mu}(x)\mathrm{d}\nu(x)} + \lambda_2 \mathrm{Ent}(\nu)$$



Minimizer

$$\boxed{p_\mu(x) \propto \exp\left(-\frac{1}{\lambda_2}\frac{\delta F(\mu)}{\delta \mu}(x)\right)}$$

$F(\mu) = \int L(x)\mathrm{d}\mu$

$\Rightarrow p_\mu \propto \exp(-\lambda_2^{-1} L(x))$

Proximal Gibbs measure

- The proximal Gibbs measure is a kind of “tentative” target.
- It plays important role in the convergence analysis.

Entropy sandwich

Proximal Gibbs measure:

$$p_\mu(x) \propto \exp\left(-\frac{1}{\lambda_2}\frac{\delta F(\mu)}{\delta\mu}(x)\right) \qquad p_\mu = \operatorname*{arg\,min}_{\nu\in\mathcal{P}} (\nu-\mu)\frac{\delta F(\mu)}{\delta\mu} + \lambda_2 \mathrm{Ent}(\nu)$$

Theorem (Entropy sandwich) [Nitanda, Wu, Suzuki (AISTATS2022)][Chizat (2022)]

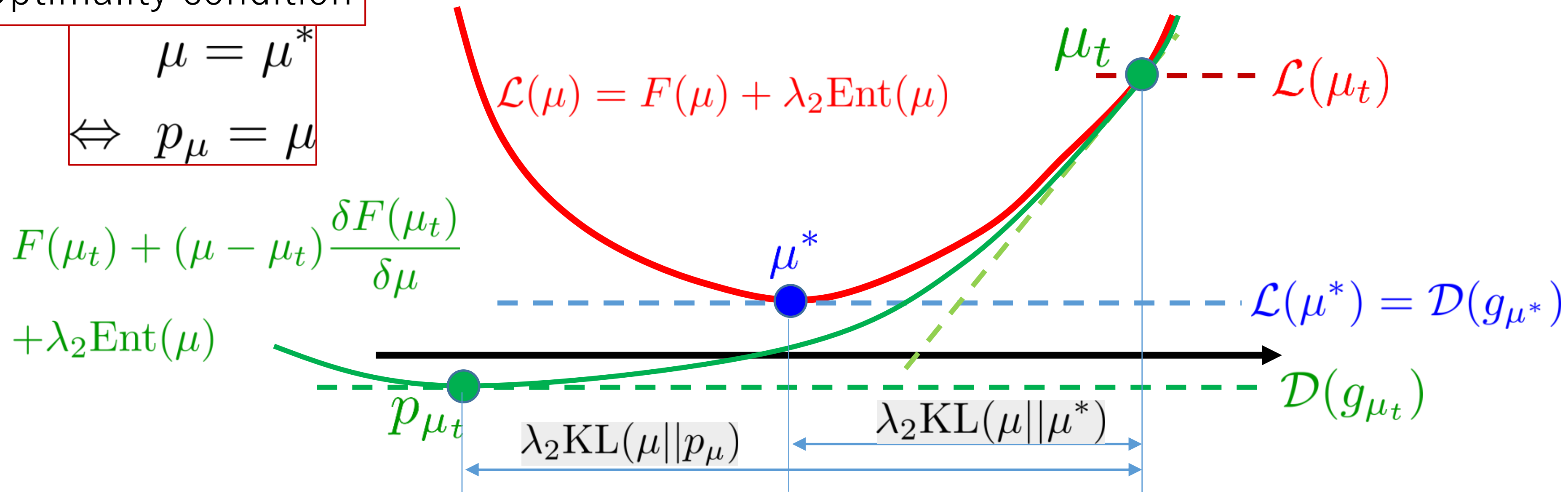
$$\mu^* = \operatorname*{arg\,min}_{\mu\in\mathcal{P}} \mathcal{L}(\mu)$$

$$\lambda_2 \mathrm{KL}(\mu\|\mu^*) = \mathcal{L}(\mu) - \underset{\substack{\| \\ \mathcal{D}(g_{\mu^*})}}{\underline{\mathcal{L}(\mu^*)}} \leq \mathcal{L}(\mu) - \mathcal{D}(g_\mu) = \lambda_2 \mathrm{KL}(\mu\|p_\mu)$$

Optimality condition

$$\mu = \mu^*$$

$$\Leftrightarrow \quad p_\mu = \mu$$



Duality (informal)

[Nitanda, Oko, Wu, Suzuki (ICML2023); Nitanda, Wu, Suzuki (AISTATS2022); Oko, Suzuki, Nitanda, Wu (ICLR2022)]

$(\mathrm{Ent}(\mu) = \int \log(\mu)\mathrm{d}\mu)$

Primal $$\min_{\mu\in\mathcal{P}} \mathcal{L}(\mu) = F(\mu) + \lambda_2 \mathrm{Ent}(\mu)$$

||

$$\min_{x\in\mathcal{X}} f(Ax) + g(x) = -\min_{g\in\mathcal{Y}^*} f^*(g) + g^*(-A^* g)$$

$A : \mathcal{X} \to \mathcal{Y}$ (bounded linear)

(Fenchel's duality theorem)
[Rockafellar (1967)]

Dual $$\max_{g:\mathbb{R}^d\to\mathbb{R}} \mathcal{D}(g) = -F^*(g) - \lambda_2 \log\left(\int \exp\left(-\frac{g(x)}{\lambda_2}\right)\mathrm{d}x\right)$$

$$F^*(g) := \sup_{\mu\in\mathcal{P}}\{\int g(x)\mathrm{d}\mu(x) - F(\mu)\}$$

Primal-Dual variable correspondence:

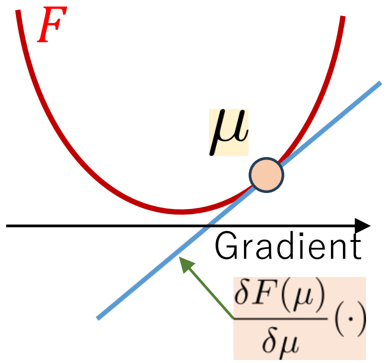
$$\overset{\text{(P)}}{\mu} \longrightarrow \overset{\text{(D)}}{g_\mu(x) := \frac{\delta F(\mu)}{\delta\mu}(x)} \longrightarrow \overset{\text{(P)}}{p_\mu(x) \propto \exp\left(-\frac{1}{\lambda_2}\frac{\delta F(\mu)}{\delta\mu}(x)\right)}$$

Duality gap and divergence:

$\mu^* = \arg\min_{\mu\in\mathcal{P}} \mathcal{L}(\mu)$

- $\mathcal{L}(\mu) - \mathcal{D}(g_\mu) = \lambda_2 \mathrm{KL}(\mu || p_\mu) \geq 0$
- $\mathcal{L}(\mu^*) = \mathcal{D}(g_{\mu^*}) \Leftrightarrow \mu^* = p_{\mu^*}$

(optimality condition)



Convergence rate

Proximal Gibbs measure:

$$p_\mu(x) \propto \exp\left(-\frac{1}{\lambda_2}\frac{\delta F(\mu)}{\delta\mu}(x)\right)$$

$$p_\mu = \arg\min_{\nu\in\mathcal{P}}(\nu-\mu)\frac{\delta F(\mu)}{\delta\mu} + \lambda_2\mathrm{Ent}(\nu)$$

Assumption (Log-Sobolev inequality)

c.f., Polyak-Lojasiewicz condition
$f(x) - f(x^*) \leq C\|\nabla f(x)\|^2$

There exists $\alpha > 0$ such that for any probability measure ν (abs. cont. w.r.t. p_μ),

$$\mathrm{KL}(\nu||p_\mu) \leq \frac{1}{2\alpha}I(\nu||p_\mu)$$

KL-div

$$\mathrm{KL}(\nu||\mu) = \int \log\left(\frac{\mathrm{d}\nu}{\mathrm{d}\mu}\right)\mathrm{d}\nu$$

Fisher-div

$$I(\nu||\mu) = \int \left\|\nabla \log\frac{\mathrm{d}\nu}{\mathrm{d}\mu}\right\|^2 \mathrm{d}\nu$$

Theorem (Linear convergence) [Nitanda, Wu, Suzuki (AISTATS2022)][Chizat (2022)]

If p_{μ_t} satisfies the LSI condition for any $t \geq 0$, then

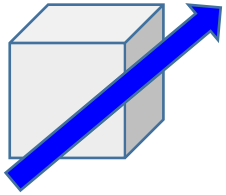
$$\mathcal{L}(\mu_t) - \mathcal{L}(\mu^*) \leq \exp(-2\alpha\lambda_2 t)(\mathcal{L}(\mu_0) - \mathcal{L}(\mu^*))$$

The rate of convergence is characterized by LSI

Proof outline of convergence

- MF-LD obeys the following nonlinear Fokker-Planck equation:

Mass: $\mu_t(x)$



Vector field: $b(x, \mu_t)$

$$\partial_t \mu_t = \lambda_2 \Delta_x \mu_t + \nabla \cdot [\mu_t \nabla \frac{\delta F(\mu_t)}{\delta \mu}]$$

$$\left(= \nabla \cdot \left[\underbrace{\left(\lambda_2 \nabla \log(\mu_t) + \nabla \frac{\delta F(\mu_t)}{\delta \mu}\right)}_{=: -v_t} \mu_t \right]\right)$$

$$= -\nabla \cdot [v_t \mu_t]$$

[Continuity equation]

Then,

$\mathcal{L}(\mu) := F(\mu) + \lambda_2 \mathrm{Ent}(\mu)$

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathcal{L}(\mu_t) = \int \left\langle v_t, \nabla \frac{\delta \mathcal{L}(\mu_t)}{\delta \mu} \right\rangle \mathrm{d}\mu_t$$

(∵continuity equation)

(Definition of p_{μ_t})

$$p_\mu(x) \propto \exp\left(-\frac{1}{\lambda_2} \frac{\delta F(\mu)}{\delta \mu}(x)\right)$$

$$= \int \langle v_t, \nabla \frac{\delta F(\mu_t)}{\delta \mu} + \lambda_2 \nabla \log(\mu_t) \rangle \mathrm{d}\mu_t$$

$$= -\int \|v_t\|^2 \mathrm{d}\mu_t = -\lambda_2^2 I(\mu_t || p_{\mu_t})$$

LSI & Entropy sandwich

$$\leq -2\alpha \lambda_2^2 \mathrm{KL}(\mu_t || p_{\mu_t}) \leq -2\alpha \lambda_2 (\mathcal{L}(\mu_t) - \mathcal{L}(\mu^*))$$

※ Since $\frac{\delta F(\mu_t)}{\delta \mu}$ nonlinearly depends on μ_t, we say “nonlinear Fokker-Planck”.

$$\text{GLD: } F(\mu) = \int L(x) \mathrm{d}\mu \Rightarrow \frac{\delta F(\mu)}{\delta \mu}(\cdot) = L(\cdot)$$

Log-Sobolev inequality

L2-regularized loss function for mean field 2-layer NN:

$$f_\mu(z) = \int h_x(z)\mathrm{d}\mu(x) \quad \text{where} \quad h_x(z) = r\sigma(w^\top z) \text{ for } x = (r, w)$$

$$F(\mu) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(f_\mu(z_i)) + \lambda_1 \mathbb{E}_\mu[\|X\|^2]$$

Proximal Gibbs:

$$p_\mu(x) \propto \exp\left(-\frac{1}{\lambda_2}\frac{\delta F(\mu)}{\delta\mu}(x)\right) = \exp\left[-\frac{1}{\lambda_2}\left(\overbrace{\underbrace{\frac{1}{n}\sum_{i=1}^{n}\ell_i'(f_\mu(z_i))h_x(z_i)}_{\text{Bounded } (\leq B)}}^{\text{Bounded perturbation}} + \overbrace{\underbrace{\lambda_1\|x\|^2}_{\text{Strongly convex}}}^{\text{Gaussian}}\right)\right]$$

If $\sup_z \left|\ell_i'(f_\mu(\cdot))h_x(\cdot)\right| \leq B$, the proximal Gibbs measure p_μ satisfies the LSI with a constant α with

$$\alpha \geq \frac{2\lambda_1}{\lambda_2}\exp(-4B/\lambda_2)$$

∵ **Bakry-Emery criterion** (1985) and **Holley-Strook bounded perturbation lemma** (1987)

Other applications

Mean field Langevin dynamics can be applied to several problems where a distribution is optimized.

- **Nonparametric density estimation via MMD minimization**

$$F(\mu) = \mathrm{MMD}^2(g * \mu, \hat{\mu}_n) + \lambda_1 \mathbb{E}_\mu[\|x\|^2]$$

k: positive definite kernel

$$\mathrm{MMD}^2(\nu_1, \nu_2) := \|k_{\nu_1} - k_{\nu_2}\|^2_{\mathcal{H}_k}$$

where $k_\mu = \int k(x,\cdot)\mu(\mathrm{d}x)$ (kernel embedding).

- ➢ $g(x) = \frac{1}{\sqrt{(2\pi\sigma^2)^d}} \exp\left(-\frac{\|x\|^2}{2\sigma^2}\right)$
- ➢ $\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}$: Empirical distribution (training data)

(see also Chizat (2022,TMLR)

- **Variational inference to approximate Bayesian posterior**

$$F(\mu) = \mathrm{KSD}(\mu) + \lambda_1 \mathbb{E}_\mu[\|x\|^2]$$

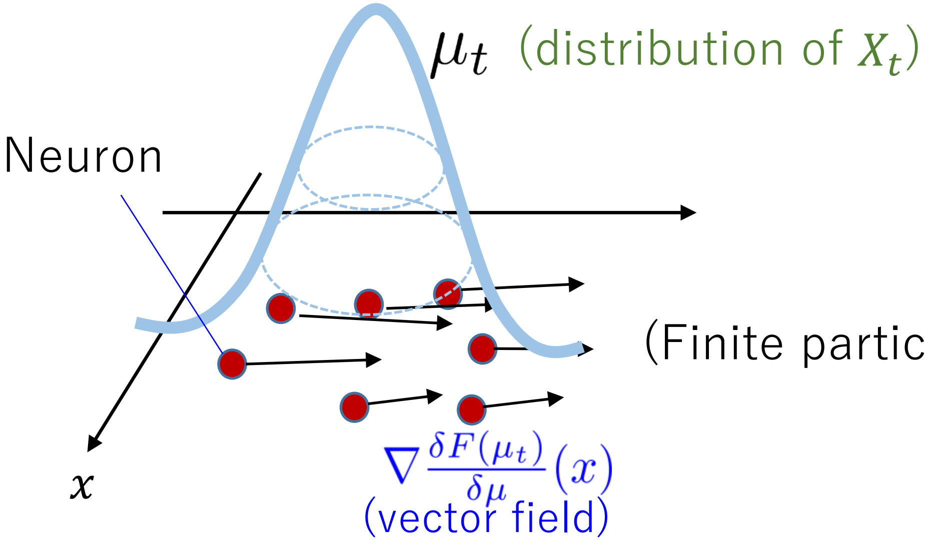
(KSD: Kernel Stein Discrepancy from a posterior distribution)

Finite particles & discrete time algorithm

We have obtained a convergence of infinite width and continuous time dynamics.

Question:

Can we evaluate a finite particles & discrete time approximation errors?



(Finite particle approximation)

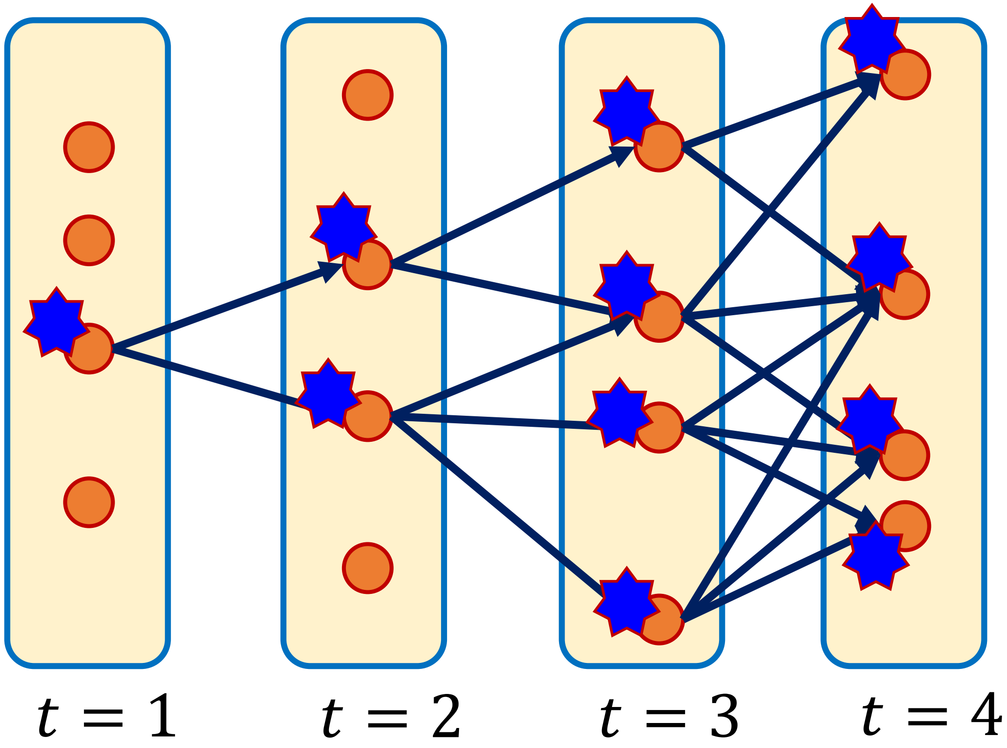
Difficulty

- SDE of interacting particles (McKean, Kac,⋯, 60’)

Propagation of chaos [Sznitman, 1991; Lacker, 2021]**:**

The particles behave as if they are independent as the number of particles increases to infinity.

Finite particle approximation error can be amplified through time.
→ It is difficult to bound the perturbation uniformly over time.



- A naïve evaluation gives exponential growth on time:

$$\exp(t)\,/M$$

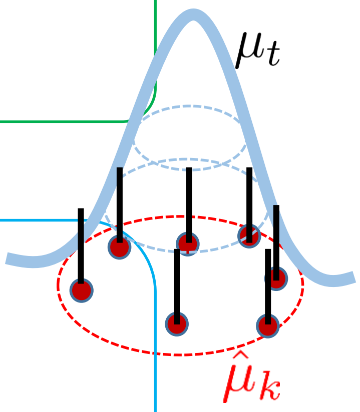
[Mei et al. (2018, Theorem 3)]

➢ Weak interaction/Strong regularization in existing work

Practical algorithm

$$\mathrm{d}X_t = -\nabla \frac{\delta F(\mu_t)}{\delta \mu}(X_t)\mathrm{d}t + \sqrt{2\lambda_2}\mathrm{d}B_t$$

M particles $\left(X_k^{(i)}\right)_{i=1}^{M}$



(time discretization)

$$X_{k+1}^{(i)} = X_k^{(i)} - \eta_k v_k^i + \sqrt{2\eta_k\lambda_2}\xi_k^{(i)}$$

where $\mathbb{E}[v_k^i] = \nabla \frac{\delta F(\hat{\mu}_k)}{\delta \mu}(X_k^i)$ and $\hat{\mu}_k = \frac{1}{M}\sum_{i=1}^{M}\delta_{X_k^{(i)}}$

(stochastic gradient) (space discretization)

➢ Noisy gradient descent on 2-layer NN with finite width.

- **Time discretization:** $t \to k\eta$ (η: step size, k: # of steps)
- **Space discretization:** μ_t is approximatd by M particles

$$\mu_t \to \hat{\mu}_k = \frac{1}{M}\Sigma\delta_{X_k^{(i)}}$$

- **Stochastic gradient:** $\nabla\frac{\delta F(\mu)}{\delta\mu} \to v_k^i$

Convergence analysis

$p_\mu(x) \propto \exp\left(-\frac{1}{\lambda_2}\frac{\delta F(\mu)}{\delta\mu}(x)\right)$: proximal Gibbs measure

Theorem (One-step update) [Suzuki, Wu, Nitanda (2023)]

Suppose that p_μ satisfies log-Sobolev inequality with a constant α.

Under smoothness and boundedness of the loss function, it holds that

$$\mathscr{L}^{(M)}(\hat{\mu}_{k+1}) - \mathcal{L}(\mu^*) \leq \exp(-\lambda_2\eta_k\alpha)\left(\mathscr{L}^{(M)}(\hat{\mu}_k) - \mathcal{L}(\mu^*)\right) + C\left(\underbrace{\eta_k^3 + \lambda_2\eta_k^2}_{\text{Time discr.}} + \underbrace{\frac{\eta_k}{M}}_{\text{Space discr.}} + \underbrace{\eta_k^{\frac{3}{2}}\lambda_2^{\frac{1}{2}}\sigma_k\tilde{\sigma}_k}_{\text{Stochastic approx.}}\right)$$

$\mathbf{O(1/M)}$

Naïve bound: $\eta_k\sigma_k^2$

$$\begin{cases} \sigma_k^2 = \max_i \mathbb{E}\left[\|v_k^i - \mathbb{E}[v_k^i]\|^2\right] \\ \tilde{\sigma}_k^2 = \max_i \mathbb{E}\left[\left\|\nabla v_k^{i\top}(\mathscr{X}) - \nabla\nabla^\top \frac{\delta F(\mu_{\mathscr{X}})}{\delta\mu}(X^i)\right\|_{\mathrm{op}}^2\right] \end{cases}$$

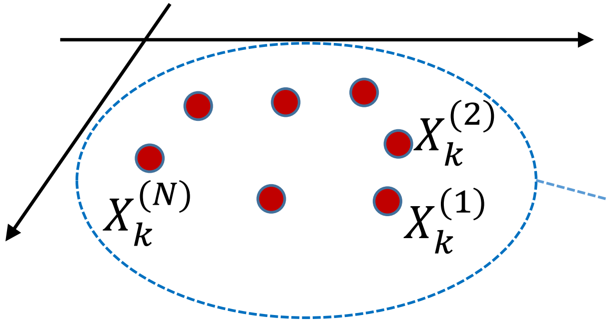
Assumption:

1. $F: \mathcal{P} \to \mathbb{R}$ is convex and has a form of $\boldsymbol{F(\mu) = L(\mu) + \lambda_1 \mathbb{E}_\mu[\|x\|^2]}$.
2. (smoothness) $\left\|\nabla\frac{\delta L(\mu)}{\delta\mu}(x) - \nabla\frac{\delta L(\nu)}{\delta\mu}(y)\right\| \leq C(W_2(\mu,\nu) + \|x-y\|)$ and (boundedness) $\left\|\nabla\frac{\delta L(\mu)}{\delta\mu}(x)\right\| \leq R$. (+ second order differentiability)

[Suzuki, Wu, Nitanda: Convergence of mean-field Langevin dynamics: Time and space discretization, stochastic gradient, and variance reduction. NeurIPS2023]

Uniform log-Sobolev inequality

Reference



$\mathcal{X}_k = \left(X_k^{(i)}\right)_{i=1}^M \sim \mu_k^{(M)}$: Joint distribution of M particles.

Potential of the joint distribution $\mu_k^{(M)}$ on $\mathbb{R}^{d\times M}$:

$$\mathscr{L}^M(\mu_k^{(M)}) = M\mathbb{E}_{\mathscr{X}\sim\mu_k^{(M)}}[F(\hat{\mu}_{\mathscr{X}})] + \lambda_2 \mathrm{Ent}(\mu_k^{(M)}).$$

where $\hat{\mu}_{\mathscr{X}} = \frac{1}{M}\sum_{i=1}^M \delta_{X^{(i)}}$ $\quad (\mathscr{X} = (X^{(i)})_{i=1}^M)$

- ➢ The finite particle dynamics is the Wasserstein gradient flow that minimizes $\mathscr{L}^M$.

(Approximate) Uniform log-Sobolev inequality [Chen et al. 2022]

For any *M*,

$$\frac{1}{M}\mathscr{L}^M(\mu_k^{(M)}) - \mathcal{L}(\mu^*) \le \frac{\lambda_2}{2\alpha}\left(\frac{1}{M}I(\mu_k^{(M)}\|p^{(M)})\right) + \frac{C_{\alpha,\lambda_2}}{M}$$

(Fisher divergence)

where $p^{(M)}(\mathscr{X}) \propto \exp(-\frac{M}{\lambda_2}F(\hat{\mu}_{\mathscr{X}}))$

Recall $\mathcal{L}(\mu) = F(\mu) + \lambda_2\mathrm{Ent}(\mu)$

[Chen, Ren, Wang. Uniform-in-time propagation of chaos for mean field Langevin dynamics. arXiv:2212.03050, 2022.]

Computational complexity

SG-MFLD

$$F(\mu) = \frac{1}{n}\sum_{j=1}^{n} \ell_j(\mu) + \lambda_1 \mathbb{E}[\|X\|^2] \qquad \text{(finite sum),}$$

$$v_k^i = \frac{1}{B}\sum_{j \in I_k} \nabla \frac{\delta \ell_j(\hat{\mu}_k)}{\delta \mu}(X_k^{(i)}) + \lambda_1 X_k^{(i)} \qquad \text{(stochastic gradient)}$$

(Mini-batch size = B)

$$\mathscr{L}^{(N)}(\hat{\mu}_k) - \mathcal{L}(\mu^*) \lesssim \exp(-\lambda_2 \eta k \alpha) + \frac{1}{\alpha\lambda_2}\left(\underbrace{\eta^2 + \lambda_2\eta}_{\text{Time discr.}} + \underbrace{\frac{1}{M}}_{\text{Space discr.}} + \underbrace{\frac{\eta + \sqrt{\eta\lambda_2}}{B}}_{\text{Stochastic approx.}}\right)$$

Iteration complexity:

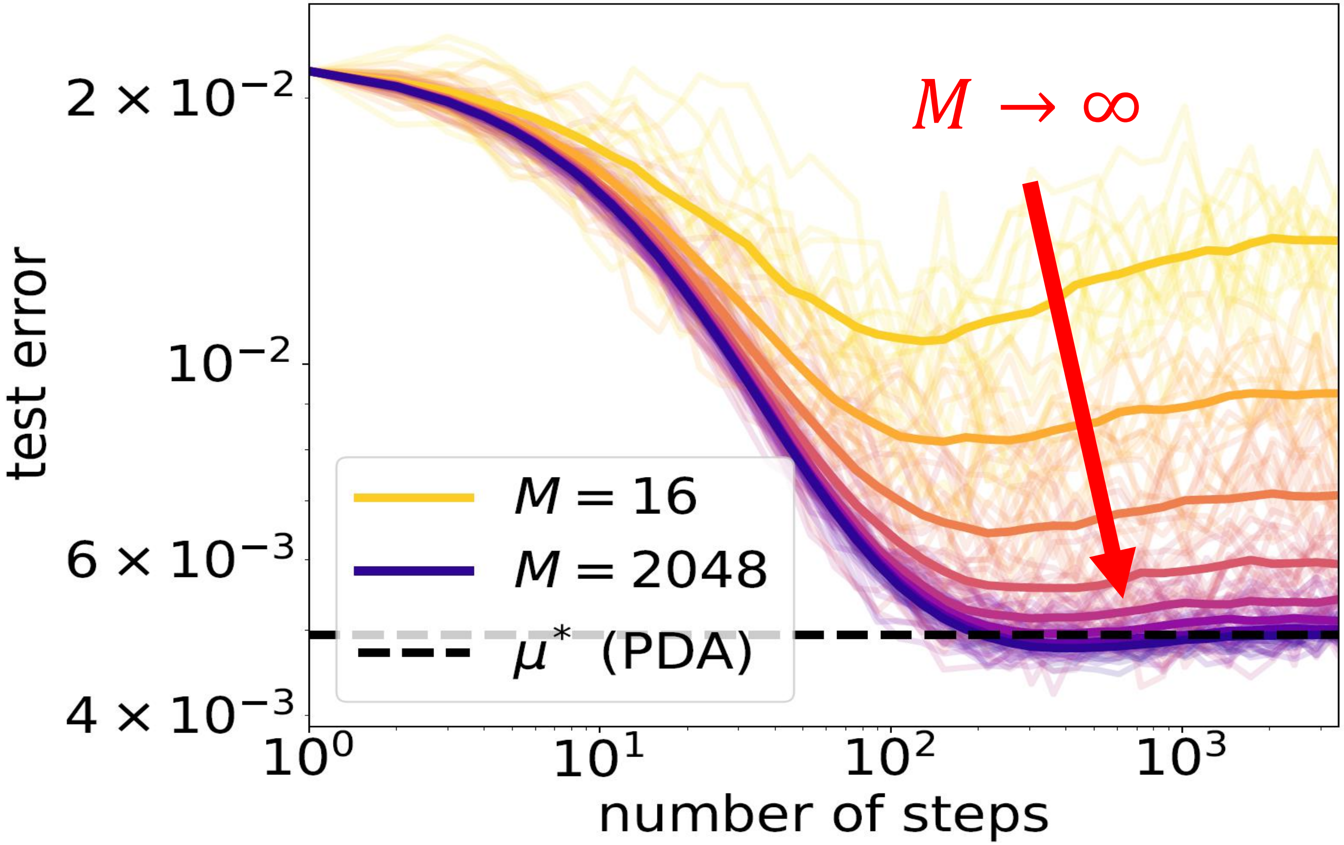
- ➢ **Approximation errors are uniform in time.**
- ➢ **No exponential dependency on M (number of**

$$k = O\left(\frac{1}{\epsilon\alpha} + \sqrt{\frac{1}{\lambda_2\epsilon\alpha}} + \left(\frac{1}{\lambda_2\epsilon\alpha}\right)^2 \frac{\lambda_2}{B^2} + \frac{1}{\lambda_2\epsilon\alpha B}\right)\frac{1}{\lambda_2\alpha}\log(\epsilon^{-1})$$

to achieve $\epsilon + O(1/(\lambda_2\alpha N))$ accuracy.

➢ $B = \sqrt{1/(\lambda_2\alpha\epsilon)}$ is the optimal mini-batch size. → $k = O(\log(\epsilon^{-1})/\epsilon)$.

Numerical experiment



Test error v.s. Number of steps
(regularization term: $r(x) = \|x\|^2$)

Generalization error analysis

Generalization error analysis

So far, we have obtained convergence of MFLD.

⇒ How effective is the feature learning of MFLD in terms of **generalization error**?

- Benefit of feature learning?

Neural network vs **Kernel method**

(NTK vs mean field)

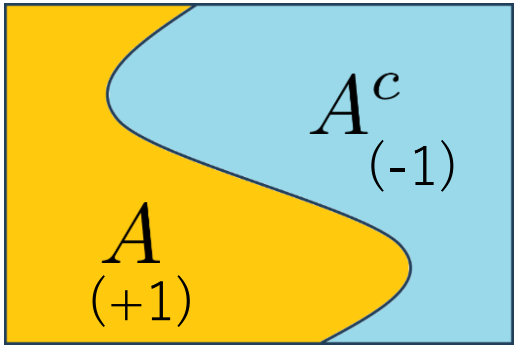
Classification task

Problem setting (classification):

$$Y = \mathbf{1}_A(Z) - \mathbf{1}_{A^c}(Z) \in \{\pm 1\}$$

$$Z \in \mathbb{R}^d$$

$$\text{Training data: } \{(z_i, y_i)\}_{i=1}^n$$



Loss function and model:

$$F(\mu) = \frac{1}{n}\sum_{i=1}^n \ell(y_i f_\mu(z_i)) + \lambda_1 \mathbb{E}_\mu[\|x\|^2]$$

$$\boxed{f_\mu(z) = \int h_x(z)\mathrm{d}\mu(x)}$$

- **Logistic loss**: $\ell(yf) = \log(1 + \exp(-yf))$
- **Tanh activation**: $h_x(z) = \bar{R} \cdot [\tanh(\langle x_1, z\rangle + x_2) + 2 \cdot \tanh(x_3)]/3$

Assumptions

Objective of MFLD:

$$\mathcal{L}(\mu) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i f_\mu(z_i)) + \lambda_1 \mathbb{E}_\mu[\|x\|^2] + \lambda \mathrm{Ent}(\mu)$$
$$= \frac{1}{n}\sum_{i=1}^{n} \ell(y_i f_\mu(z_i)) + \lambda \underline{\mathrm{KL}(\nu, \mu)}$$

where $\nu = N(0, \lambda/(2\lambda_1))$.

KL-regularization

Assumption

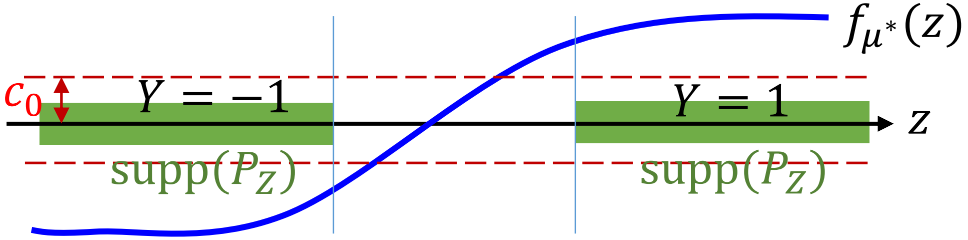
There exists μ^* such that

1. $\mathrm{KL}(\nu, \mu^*) \le R$,
2. $Y f_{\mu^*}(Z) \ge c_0 \quad (\mathrm{a.s.})$,

for some constants $R, c_0 > 0$.

(+ classification calibration condition)

The Bayes classifier is attained by μ^* with a bounded KL-div from ν.



Main theorem

Theorem 1

Suppose that $\lambda = \Theta(1/R)$, then it holds that

$$P(Yf_{\hat{\mu}}(Z) \leq 0) \lesssim \left[\frac{\bar{R}^2 R}{n}(1 + t + \log(\log(n))) + \frac{\bar{R} + 1/(Rn)}{n}\right]$$

Class. error

$$\mathrm{O}\left(\frac{\bar{R}^2 R}{n}\right)$$

with probability $1 - \exp(-t)$.

Existing bound: Chen et al. (2020); Nitanda, Wu, Suzuki (2021)

Class. Error $\leq \mathrm{O}\left(\frac{1}{\sqrt{n}}\right)$. (Rademacher complexity bound)

- Our bound provides *fast learning rate* (faster than $1/\sqrt{n}$).

$$\mathrm{O}(R/n) \ll \mathrm{O}(1/\sqrt{n})$$

$$\mathcal{L}(\mu) = \frac{1}{n}\sum_{j=1}^{n} \ell(y_i f_\mu(z_i)) + \lambda \mathrm{KL}(\nu, \mu)$$

- μ^*: $\mathrm{KL}(\nu, \mu^*) \leq R$, $\; Yf_{\mu^*}(Z) \geq c_0$
- $h_x(z) = \bar{R} \cdot [\tanh(\langle x_1, z\rangle + x_2) + 2 \cdot \tanh(x_3)]/3$

Main theorem 2

Theorem 2

Suppose that $\lambda = \overset{(\lambda = c_0/R)}{\Theta(1/R)}$ and

$$\frac{\bar{R}^3 R^2}{n}(\lambda + \bar{R}) \lesssim c_0$$

then it holds that

$$P(Y f_{\hat{\mu}}(Z) \leq 0) = 0 \text{ with probability } 1 - \exp\left(-\frac{c_0^3 n / R^2}{2\bar{R}^4}\right)$$

Theorem 1: $\mathbb{E}[\text{Class. Error}] \leq O\left(\frac{R}{n}\right)$.

Theorem 2: $\mathbb{E}[\text{Class. Error}] \leq O(\exp(-O(n/R^2)))$ if $\underline{n \geq R^2}$.

If we have sufficiently large training data,
we have exponential convergence of test error.

We only need to evaluate R to obtain a test error bound.

Proof sketch

$$\bar{F}(\mu) := \mathbb{E}_{X,Y}[\ell(Y f_\mu(X))]$$

Expected loss

$$\mu^\circ = \arg\min_{\mu \in \mathcal{P}} \left\{ \bar{F}(\mu) + \lambda \mathrm{KL}(\nu, \mu) \right\}$$

Local Rademacher complexity yields the following bound:

$$\underbrace{\bar{F}(\hat{\mu}) - \bar{F}(\mu^\circ) - (\hat{\mu} - \mu^\circ)\frac{\delta \bar{F}(\mu^\circ)}{\delta \mu}}_{\text{(I)}} + \underbrace{\lambda \mathrm{KL}(\mu^\circ, \hat{\mu})}_{\text{(II)}} \lesssim \sqrt{\frac{\mathrm{KL}(\mu^\circ, \hat{\mu})}{n}}$$

Adaptive to the KL-divergence

$$\leq \frac{1}{2n\lambda} + \frac{\lambda}{2}\mathrm{KL}(\mu^\circ, \hat{\mu})$$

By setting $\lambda = O(1/R)$,

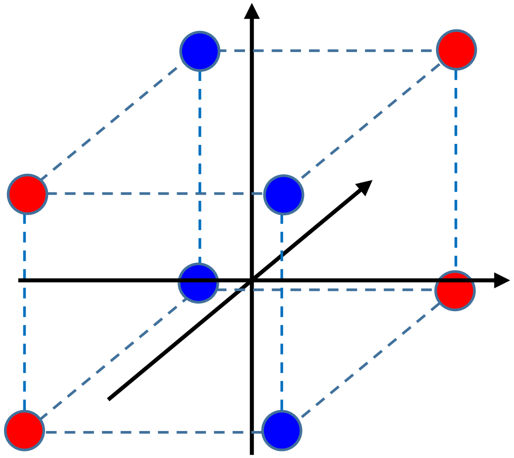
- The term (I) gives the first bound: $O(R/n)$.
- The term (II) gives the second bound: $O(\exp(-O(n/R^2)))$.

Example: k-sparse parity problem

- k-sparse parity problem on high dimensional data
  - $Z \sim \text{Unif}(\{-1,1\}^d)$ (up to freedom of rotation)
  - $Y = \prod_{j=1}^{k} Z_j$

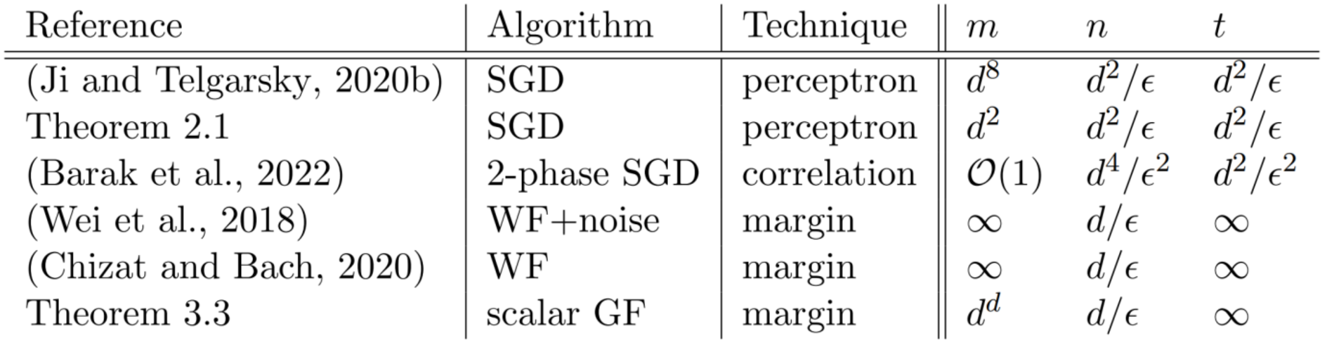
※ Assume we don't know which coordinate corresponds to Z_j.

Q: Can we learn sparse k-parity with GD?
Is there any benefit of neural network?



$k = 2$: XOR problem
$d = 3, k = 2$

Complexity to learn XOR function ($k = 2$)



Reference	Algorithm	Technique	m	n	t
(Ji and Telgarsky, 2020b)	SGD	perceptron	d^8	d^2/ϵ	d^2/ϵ
Theorem 2.1	SGD	perceptron	d^2	d^2/ϵ	d^2/ϵ
(Barak et al., 2022)	2-phase SGD	correlation	$\mathcal{O}(1)$	d^4/ϵ^2	d^2/ϵ^2
(Wei et al., 2018)	WF+noise	margin	∞	d/ϵ	∞
(Chizat and Bach, 2020)	WF	margin	∞	d/ϵ	∞
Theorem 3.3	scalar GF	margin	d^d	d/ϵ	∞

Table 1 of [Telgarsky: Feature selection and low test error in shallow low-rotation ReLu networks,

Generalization bound

Reminder

Suppose that there exists μ^* such that

$$\mu^*: \mathrm{KL}(\nu, \mu^*) \leq R,\ Y f_{\mu^*}(Z) \geq c_0 \text{ (perfect classifier with margin } c_0\text{)}$$

Then,

Theorem 1: $\mathbb{E}[\text{Class. Error}] \leq \mathrm{O}\left(\frac{R}{n}\right)$.

Theorem 2: $\mathbb{E}[\text{Class. Error}] \leq \mathrm{O}(\exp(-\mathrm{O}(n/R^2)))$ if $n \geq R^2$.

We can evaluate R required for the k-sparse parity problem:

Lemma

For the k-parity problem, we may take

$$R = \mathcal{O}\left(k \log(k) d\right)$$

Generaliza

Our analysis provides
- better sample complexity
- discrete-time/finite-width analysi
- d and k are "decoupled."

Corollary (Test accuracy of MFLD)

- Setting 1: $n > d$
  - ➢ Test error (classification error) = $\mathbf{O}(d/n)$
- Setting 2: $n > d^2$
  - ➢ Test error (classification error) = $\mathbf{O}(\exp(-n/d^2))$

(Computational complexity is $\exp(O(d))$ (But, can be relaxed to O(1) if X is anisotrop

These are better than NTK (kernel method);
Sample complexity of NTK $n = \Omega(d^k)$ vs NN $n = O(d)$

Trade-off between computational complexity and sample complexity.

Authors	regime/method	k-parity	class error	width	# iterations
Ji and Telgarsky (2019)	NTK/SGD	No	d^2/n	d^8	d^2/ϵ
Telgarsky (2023)	NTK/SGD	No	d^2/n	d^2	d^2/ϵ
Barak et al. (2022)	Two phase SGD	Yes	$d^{(k+1)/2}/\sqrt{n}$	$O(1)$	d/ϵ^2
Wei et al. (2019)	mean-field/GF	No	d/n	∞	∞
Telgarsky (2023)	mean-field/GF	No	d/n	d^d	∞
Ours	mean-field/MFLD	Yes	$\exp(-O(n/d^2))$	$e^{O(d)}$	$e^{O(d)}$
Ours	mean-field/MFLD	Yes	d/n	$e^{O(d)}$	$e^{O(d)}$

Discussion

- The CSQ lower bound states that $O(d^{k-1})$ sample complexity is optimal for methods with polynomial order computational complexity. [Abbe et al. (2023); Refinetti et al. (2021); Ben Arous et al. (2022); Damian et al. (2022)]
- On the other hand, our analysis is about full-batch GD.

	Minibatch size	# of iterations	Sample complexity
Our analysis	$\boldsymbol{n}$	$\boldsymbol{e^d}$	$\boldsymbol{d}$
SGD (CSQ-lower bound)	1	d^{k-1}	d^{k-1}

We obtain a better sample complexity than $O(d^{k-1})$ with higher computational complexity.
→ We can obtain a polynomial order method with MFLD for anisotropic input.

CSQ algorithm

Def (Correlational Statistical Query (CSQ) algorithm)

[Ben-David, Itai, Kushilevitz, 1995; Kearns, 1998; Bshouty, Feldman, 2002]

A CSQ algorithm can access the data only via queries $\phi: \mathbb{R}^d \to [-1,1]$ and returns $g \in \mathbb{R}^d$ with tolerance τ such that

$$g \in \mathbb{E}_{Z,Y}[\phi(X)Y] + [-\tau, \tau]$$

Ex. Online SGD for a squared loss:

$$\mathbb{E}_Z[(f^\circ(Z) - f_{\mathscr{X}}(Z))^2] \quad \Rightarrow \quad f^\circ(z_i)\nabla f_{\mathscr{X}}(z_i) \quad \text{(CSQ)}$$

- Boolean case: $z \sim \mathrm{Unif}(\{-1,+1\}^d)$

k-parity: $f^\circ(z) = \prod_{j \in S} z_j$ where $|S| = k$.

- Gaussian case: $x \sim N(0, I)$

Single index model: $f^\circ(x) = g(w^\top x)$

where $g: \mathbb{R} \to \mathbb{R}$ and $w \in \mathbb{R}^d$.

CSQ lower bound

For the Gaussian single index model, the information exponent plays an important role.

Hermite polynomial expansion of the link function:

$$g(z) = \sum_{k=1}^{\infty} \alpha_k h_k(z)$$

Def (information exponent [Ben Arous, Gheissari, Jagannath, 2021])

$$k^* := \arg\min_k \{k \mid \alpha_k \neq 0\}$$

The computational complexity of a CSQ algorithm is lower bounded as:

Theorem (CSQ lower bound [Abbe, Boix-Adser`, Misiakiewicz, 2023])

A CSQ algorithm with error tolerance τ requires at least N queries to obtain an estimator $\hat{f}$ s.t. $\mathbb{E}\left[\left(\hat{f} - f^\circ\right)^2\right] \leq 0.1$ where

$$N/\tau^2 \geq \begin{cases} d^k & \text{(Boolean case)}, \\ d^{k^*/2} & \text{(Gaussian case)}. \end{cases}$$

(we suppose $k^* > 2$)

Note that the gradient computation at each iteration consumes $O(d)$ queries. Thus, d^{k-1} iterations are enough.

Recent progress

- SGD with smoothing operation achieves the Gaussian optimal rate:

  Damien et al.: Smoothing the Landscape Boosts the Signal for SGD Optimal Sample Complexity for Learning Single Index Models. NeurIPS2023.

- Near optimal complexity of SGD to learn XOR problem:

  Glasgow: SGD Finds then Tunes Features in Two-Layer Neural Networks with near-Optimal Sample Complexity: A Case Study in the XOR problem. ICML2024.

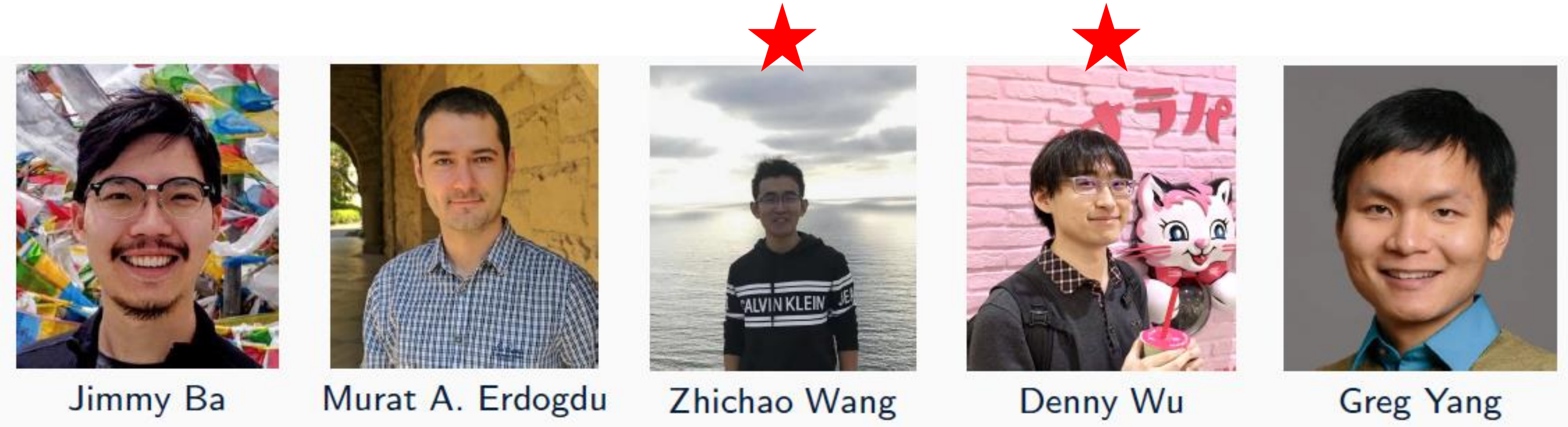
- Optimal SQ sample complexity to learn Gaussian single index model with the “generative” information exponent:

  Damian, Pillaud-Vivien, Lee, Bruna: The Computational Complexity of Learning Gaussian Single-Index Models. arXiv:2403.05529.

The setting of $k^*=1$.

Feature learning with one-step gradient descent

[Ba, Erdogdu, Suzuki, Wang, Wu, Yang: High-dimensional Asymptotics of Feature Learning: How One Gradient Step Improves the Representation. NeurIPS2022]



Gradient descent and kernel alignment

$$f_{\mathrm{NN}}(x) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} a_i \sigma(\langle x, w_i \rangle) = \frac{1}{\sqrt{N}} a^\top \sigma(W^\top x)$$

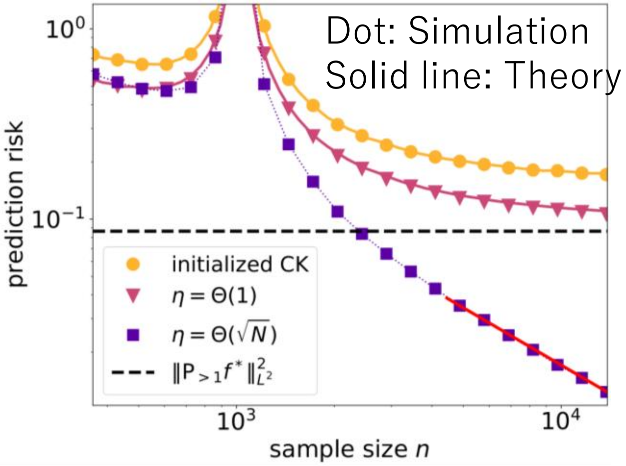
Question: Can we obtain “good” features from data by updating the first layer parameter W by gradient descent?
Result: GD with large step size can extract the leading term of the true function. Especially, for the single index model $(f^*(x) = \sigma^*(\langle x, w^* \rangle))$, the **predictive risk provably outperforms random feature methods**.
→ Kernel alignment, feature learning.

$$W_{k+1} = W_k - \eta \sqrt{N} \nabla_W L(f_{\mathrm{NN}})$$

We consider the **proportional limit** $(\boldsymbol{n}, \boldsymbol{d}, \boldsymbol{N} \to \infty)$, and evaluate predictive risk of **one-step GD.**

- ➢ $\boldsymbol{\eta = \Theta(\sqrt{N})}$ can outperform random feature models.
- ➢ $\boldsymbol{\eta = \Theta(1)}$ can outperform the initial setting of W.
- ➢ $\boldsymbol{\eta = o(1)}$ does not improve the performance.

Gaussian equivalence property + Random matrix theory
→ Exact risk evaluation.



Dot: Simulation
Solid line: Theory

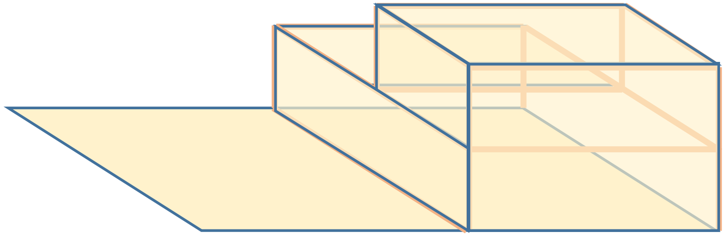
Related work

The first few step of GD with large learning rate can extract informative features.

- Staircase function

  [Abbe et al., NeurIPS2021; Abbe et al., arXiv2202.08658]

  Small number of gradient descent can extract nonlinear features to estimate “staircase” function. The trained features for GD can outperform random feature model.



- Benign overfitting with feature learning

  [Cao et al., arXiv:2202.06526; Frei et al., arXiv:2202.05928]

  Gradient descent in two-layer NN can yield benign overfitting and achieves almost the Bayes error in binary classification.

Problem setting

Observation model:

$$y_i = f^*(x_i) + \epsilon_i \quad (i = 1, \ldots, n)$$

where $x_i \sim N(0, I), \epsilon_i \sim N(0,1)$, and $x_i \in \mathbf{R}^d$.

- We fit 2-layer NN of mean field scaling:

$(\because a_i = O_p(1/\sqrt{N}))$

$$f_{\mathrm{NN}}(x) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} a_i \sigma(\langle x, w_i \rangle) = \overbrace{\frac{1}{\sqrt{N}}}^{\text{Mean field regime } O(1/N)} a^\top \sigma(W^\top x)$$

where $a_i \sim N(0, \underset{\text{var}}{1/N})$ and $W_{ij} \sim N(0, \underset{\text{var}}{1/d})$.

Empirical risk:

$$\mathcal{L}(f) = \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2$$

Predictive risk:

$$\mathcal{R}(f) = \mathbb{E}[(f^*(X) - f(X))^2]$$

Question: Can we provably improve the predictive risk by gradient descer

We analyze the risk especially for the single index model:

$$f^*(x) = \sigma^*(\langle x, \beta^* \rangle)$$

Feature learning with optimization guarantee

$$f_{\mathrm{NN}}(x) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} a_i \sigma(\langle x, w_i \rangle) = \frac{1}{\sqrt{N}} a^\top \sigma(W^\top x)$$

$$W_{k+1} = W_k - \eta \sqrt{N} \nabla_W L(f_{\mathrm{NN}})$$

We consider the **proportional limit $(n, d, N \to \infty$ with $n/d \to \psi_1, N/d \to \psi_2)$.**

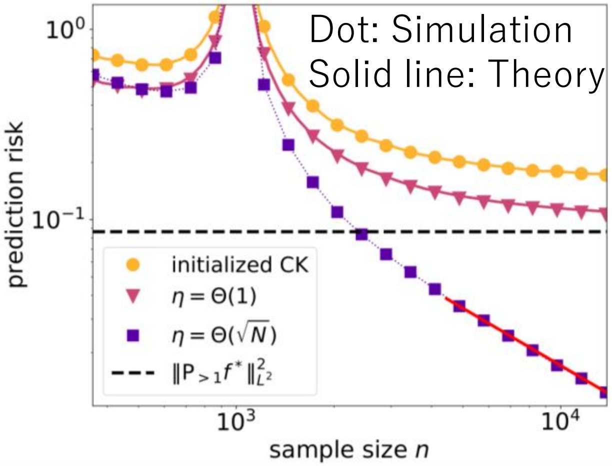
It allows to derive precise risk.

We evaluate predictive risk of **one-step GD.**

Take home message:
GD with Large step-size can outperform **any** random feature model by only one-step update.

[Outline of our result]

- $\boldsymbol{\eta = \Theta(\sqrt{N})}$ can get out of NTK regime and outperform random feature models.
- $\boldsymbol{\eta = \Theta(1)}$ can outperform the initial setting of W.
- $\boldsymbol{\eta = o(1)}$ does not improve the performance.



Dot: Simulation
Solid line: Theory

Ridge regression with RF

Feature learning vs Random feature

Random features (without feature learning):

- **Conjugate kernel at initialization:**

$$\phi_{\mathrm{CK}}(x) = \frac{1}{\sqrt{N}} \sigma(W_0^\top x)$$

- **NTK** (Neural tangent kernel):

$$\phi_{\mathrm{NTK}}(x) = \frac{1}{\sqrt{Nd}} \mathrm{Vec}(\sigma'(W_0^\top x) x^\top)$$

Precise asymptotics has been extensively studied. (e.g., [Louart, Liao, and Couillet, 2018; Mei and Montanari, 2019])

$$\hat{a}_{\mathrm{RF}} = \mathop{\arg\min}_{a \in \mathbb{R}^N} \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_i - \langle a, \phi_{\mathrm{RF}}(x_i) \rangle)^2 + \frac{\lambda}{N} \|a\|^2 \right\}$$

RF ∈ {CK,NTK}

Trained feature:

$$\phi_{\mathrm{CK}^{(t)}}(x) = \frac{1}{\sqrt{N}} \sigma(W_t^\top x)$$

Limitation of RF

(1) Random feature models and
(2) GD updates with <u>small learning rate</u>
can learn only <u>linear functions</u> in the proportional

[El Karoui (2010); Ghorbani et al. (2019), Hu and Lu (2020),…] $\mathcal{R}_{XX}(f) = \mathbb{E}[(f^*(X) - \hat{f}_{XX}(X))^2]$

Theorem (Lower bound of predictive risk for RF)

If the step size is not large $\eta = \Theta(1)$, then for any finite number steps t, we have

$$\inf_{\lambda>0} \min\{\mathcal{R}_{\mathrm{CK}}(\lambda), \mathcal{R}_{\mathrm{NTK}}(\lambda), \mathcal{R}_{\mathrm{CK}^{(t)}}(\lambda)\} \geq \|P_{>1} f^*\|_{L^2(P_X)}^2 + o_{p,d}(1)$$

Nonlinear part cannot be trained!

$$P_{>1} f^* := (I - P_{\leq 1}) f^*$$

where $P_{\leq 1}$ is the projection operator in $L^2(P_X)$ to the subspace consisting of linear functions and constants.

Remark: The same is true for “rotational invariant kernel” [El Karoui (2010)].

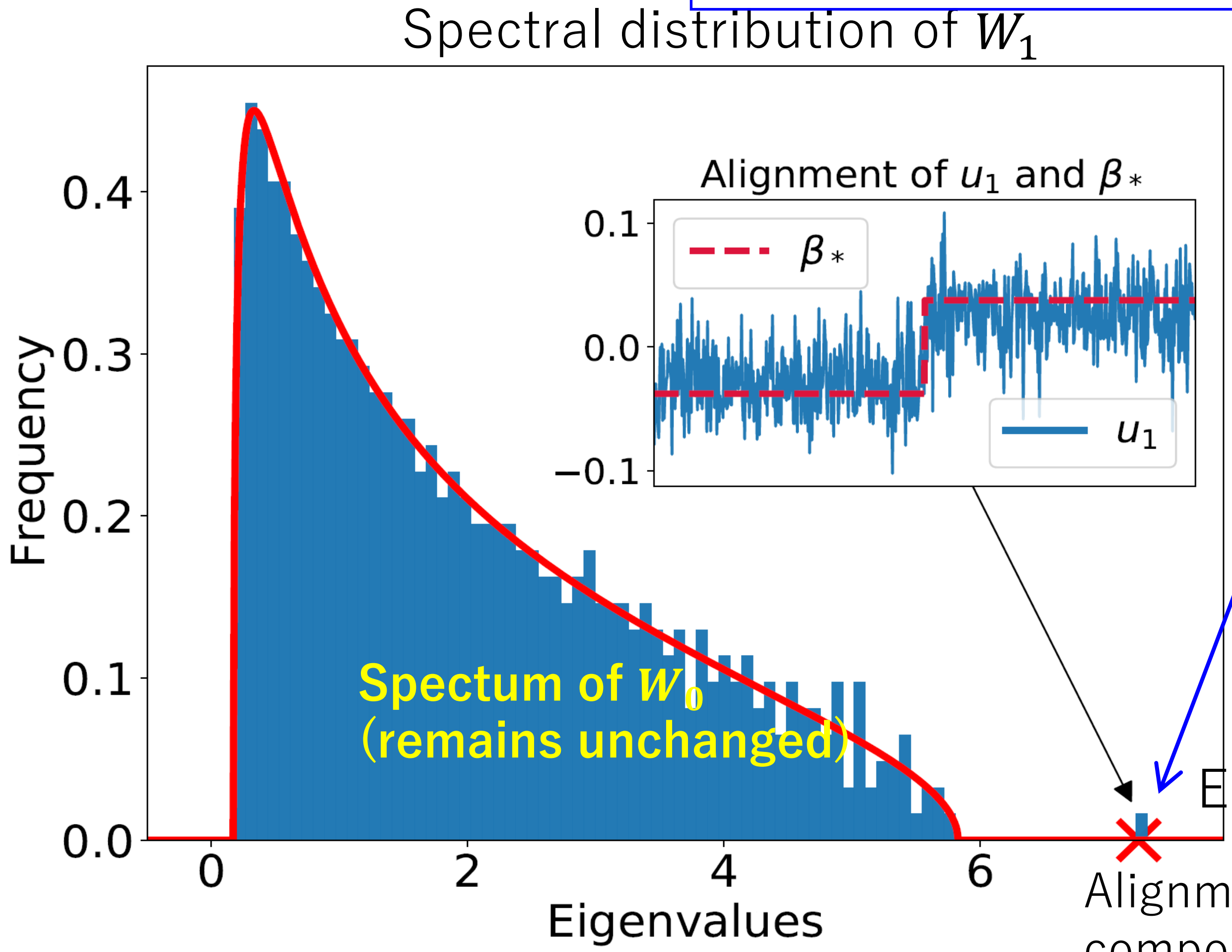
This is because in high dimensional setting, a central limit theorem yields

$$a^\top \phi_{\mathrm{CK}}(x) = \frac{1}{\sqrt{N}} a^\top \sigma(W_0^\top x_i) \approx \frac{1}{\sqrt{N}} a^\top (\mu_1 W_0^\top x_i + \mu_2 z)$$

(linear function;
Gaussian equivalence)

Effect of large step-size update

$$W_1 = W_0 + \eta\sqrt{N}\underbrace{(-\nabla_W \mathcal{L}(f_{\mathrm{NN}}^{(0)})/2)}_{\boxed{\textbf{Theorem: almost rank 1}}}$$

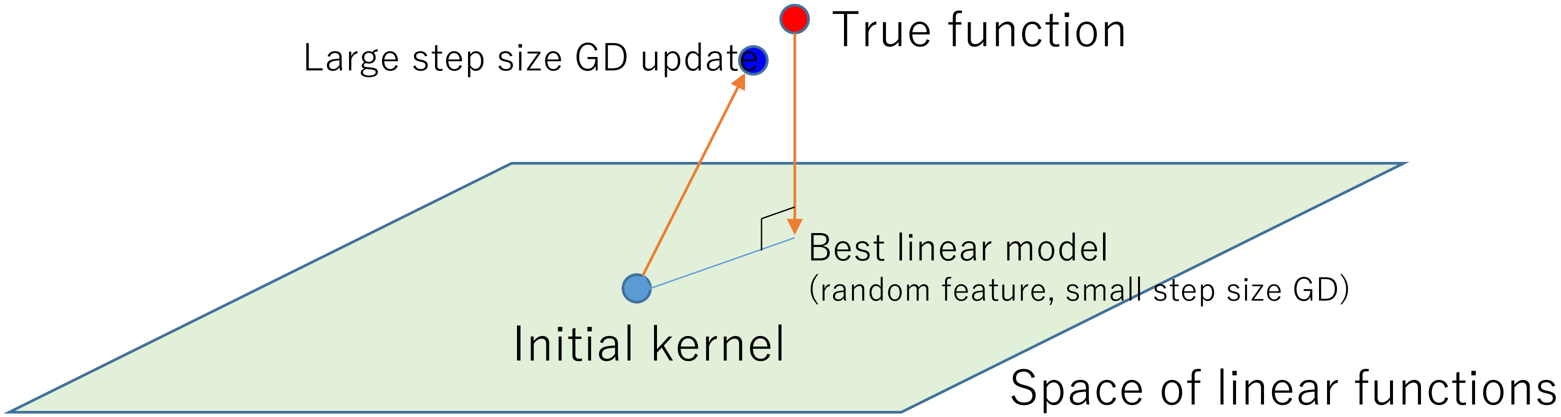


Spectum of W_0 (remains unchanged)

Estimated signal

Alignment with the linear component of f^*.
(feature learning)

Improvement over the Initial CK



- $\eta = \Theta(\sqrt{N})$ (large learning rate):

Known as maximal update parameterization $(\mu\mathrm{P})$ [Yang and Hu, 2020].

$$\tau^* = \inf_{\eta>0} \mathbb{E}_{\xi_1\sim N(0,1)} \left[\sigma^*(\xi_1) - \mathbb{E}_{\xi_2\sim N(0,1)}[\sigma(\eta\xi_1+\xi_2)]\right]$$

(measure for model misspecification)

$f^*(x) = \sigma^*(\langle x, \beta^* \rangle)$ is assumed.

- $\tau^* = 0$ if $\sigma = \sigma^* = \mathrm{erf}$.
- $\tau^* \ll 1$ if $\sigma = \sigma^* = \tanh$.

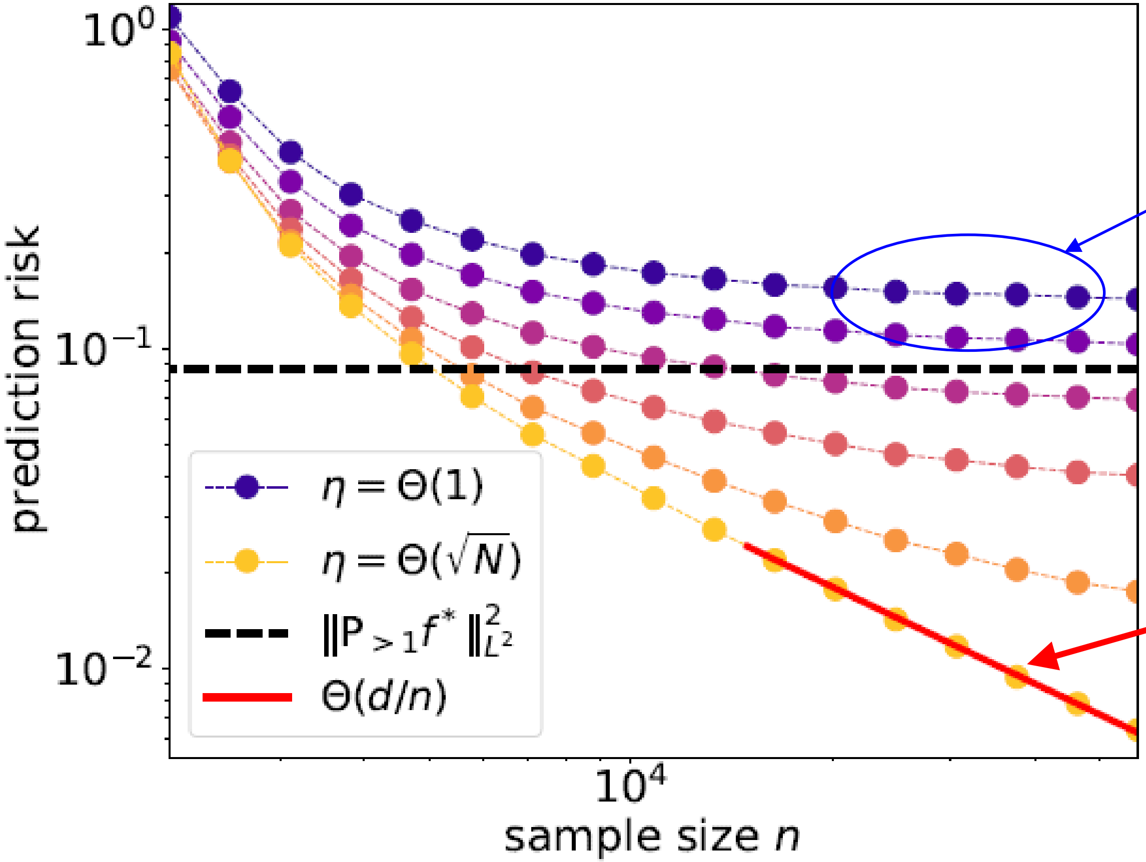
$$\boxed{\mathcal{R}_{W_1}(\lambda) \le 16\tau^* + C(\sqrt{\tau^*}\psi_1^{-1/2} + \psi_1^{-1}) + o_p(1)}$$

$n/d \to \psi_1,\ N/d \to \psi_2$

Large learning rate yields feature learning and can be better than the small step size regime if $\tau^* \ll \|P_{>1} f^*\|^2$.

Implications

Corollary If $\sigma = \sigma^* = \mathrm{erf}$, then $\tau^* = 0$.
In particular, we have $R_{W_1}(\lambda) = \Theta(\psi_1^{-1}) = \Theta(d/n)$.



Small step size

Lower bound for RF models

$\Theta(d/n)$

Outperform any RF model

Predictive risk of ridge regression on CK obtained by one step GD (empirical simulation, $d = 1024$): brighter color represents larger step size scaled as $\eta = N^\alpha$ for $\alpha \in [0,1/2]$. We chose $\sigma = \sigma^* = \mathrm{erf}$, $\psi_2 = 2, \lambda = 10^{-3}$, and $\sigma_\epsilon = 0.1$.

Summary

- **Representation/Generalization ability**
  - Depth separation
  - Adaptivity of deep learning: separation between linear (shallow) and deep methods
- **Optimization ability**
  - Overparameterization
  - Noisy gradient descent: a near global optimum
    - ✓ Estimation error separation between kernel and deep learning
  - Mean field Langevin
  - CSQ lower boun

Deep learning theory that makes DL white box that can be *controllable*.

It would reveal the essence of "good learning system" which would be useful to develop methods *beyond* DL.