Estimation ability of deep learning with connection to sparse estimation in function space

Taiji Suzuki

The University of Tokyo

AIP-RIKEN

Collaboration with Satoshi Hayakawa, Atsushi Nitanda, Kenta Oono.

5th/Nov/2019

4TU.AMI Meeting on Mathematics of Deep Learning

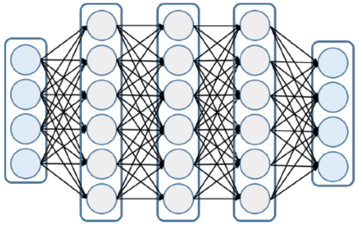
@TU Delft, Science Centre

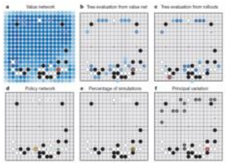
Theory of Deep Learning

Deep learning

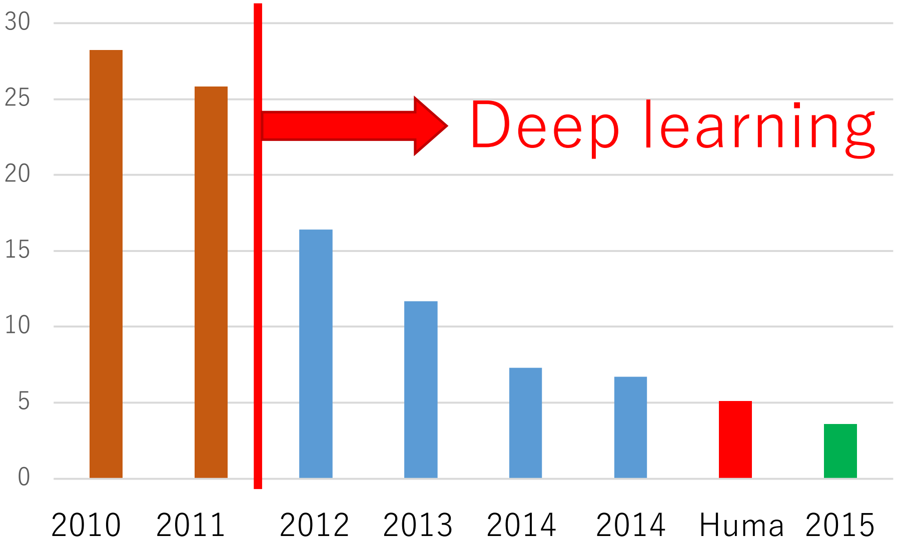
- High performance
- Applied to services in several industries:

Google Deepmind, Facebook AI Lab., Baidu, ...

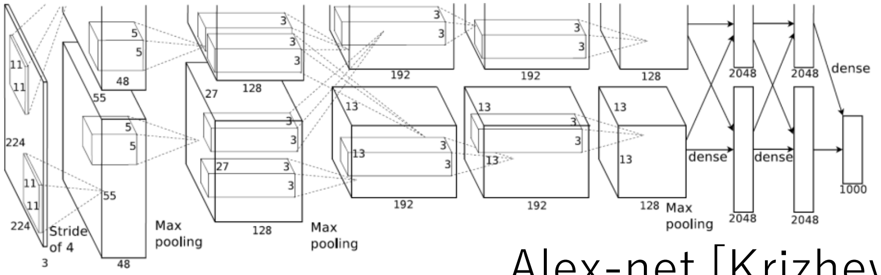




- **High performance in several applications**
- **But, theoretical understanding is not satisfactory (Big issue all over the world)**

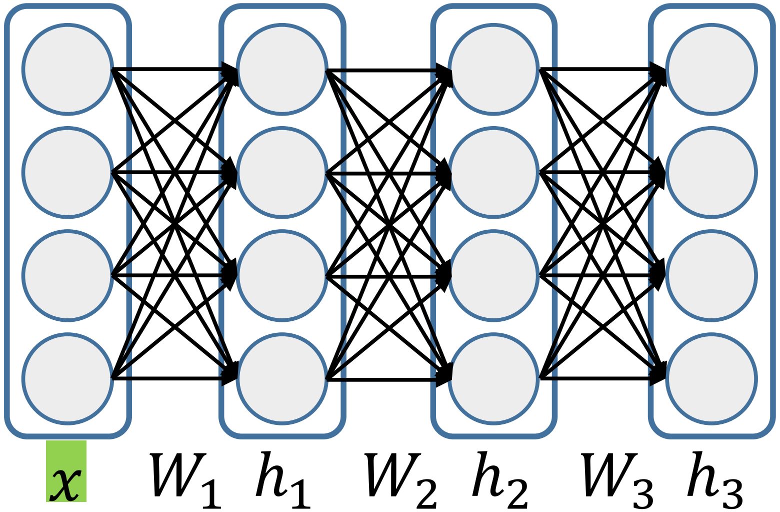


Result of ILSVRC
(classification error (%))

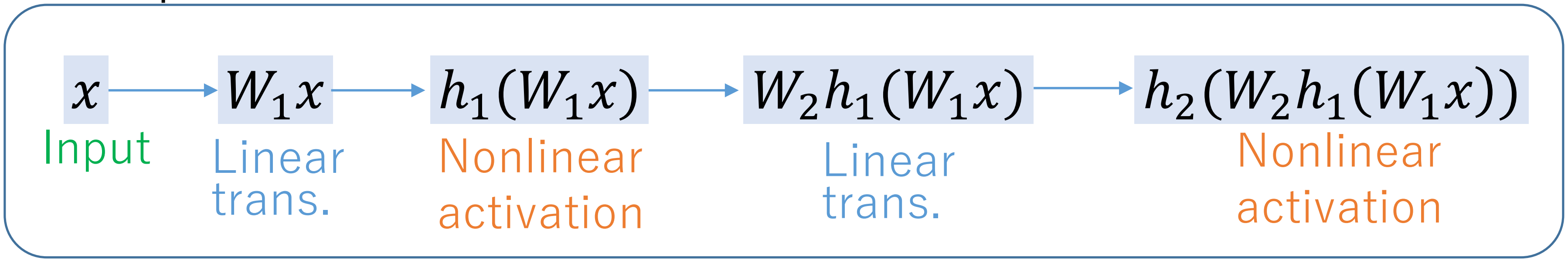


Alex-net [Krizhevsky, Sutskever + Hinton, 2012]

Structure of deep NN



Repeat “linear transform” and “nonlinear activation.”



$x \rightarrow W_1 x \rightarrow h_1(W_1 x) \rightarrow W_2 h_1(W_1 x) \rightarrow h_2(W_2 h_1(W_1 x))$

Input

Linear trans.

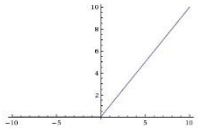
Nonlinear activation

Linear trans.

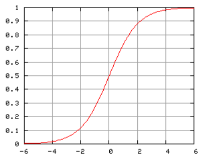
Nonlinear activation

$h_1(u) = [h_{11}(u_1), h_{12}(u_2), \ldots, h_{1d}(u_d)]^T$

- ☆ReLU (Rectified Linear Unit)： $h(u) = \max\{u, 0\}$



- Sigmoid function： $h(u) = \dfrac{1}{1 + e^{-u}}$

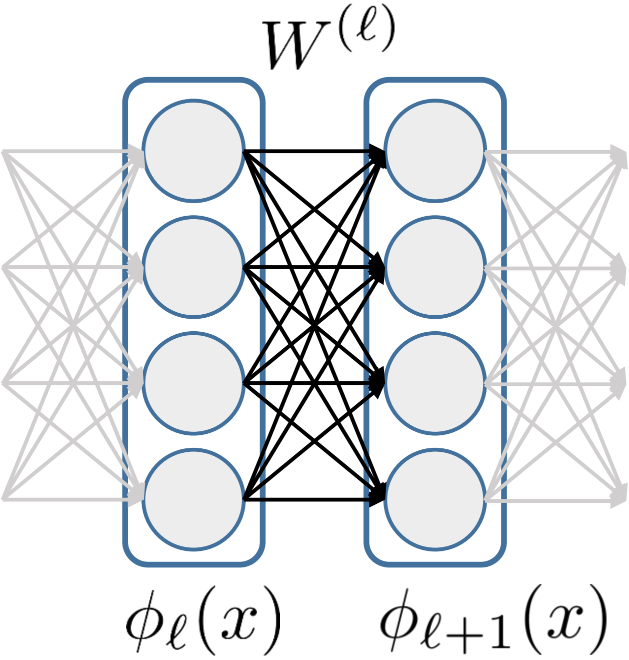


Fully connected layer

- ℓ -th layer

$$\phi_{\ell+1}(x) = \eta(W^{(\ell)}\phi_\ell(x) + b^{(\ell)})$$

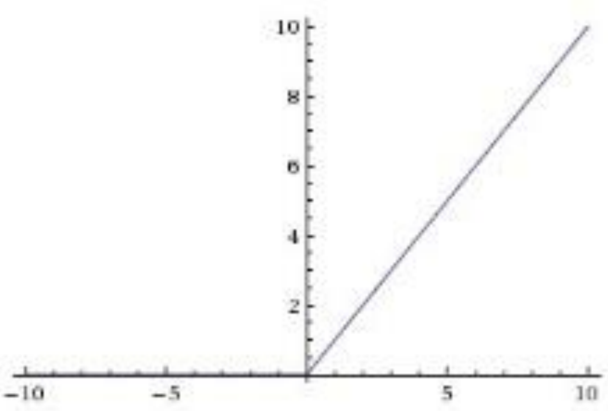
$$W^{(\ell)} \in \mathbb{R}^{m_{\ell+1}\times m_\ell} \quad b^{(\ell)} \in \mathbb{R}^{m_{\ell+1}}$$



Examples of activation functions

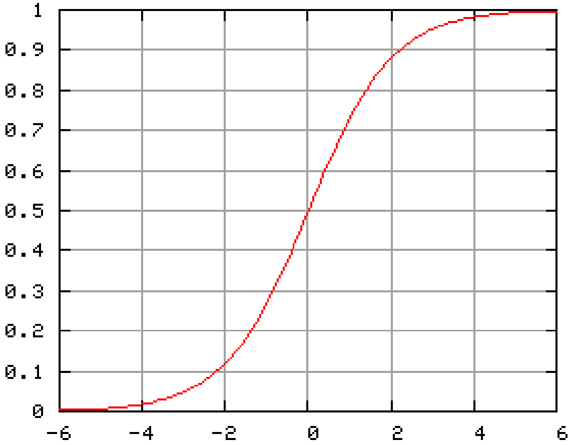
☆ReLU (Rectified Linear Unit)

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



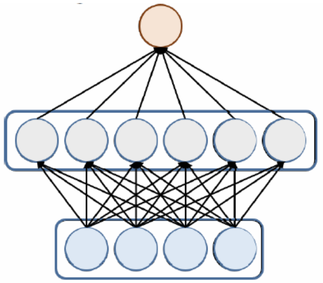
Sigmoid function

$$\eta(u) = \frac{1}{1 + e^{-u}}$$



Universal Approximator

$$f(x) = \sum_{j=1}^{m} v_j \eta(w_j^\top x + b_j)$$

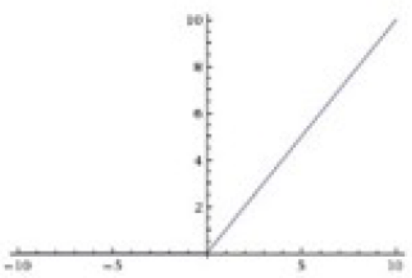


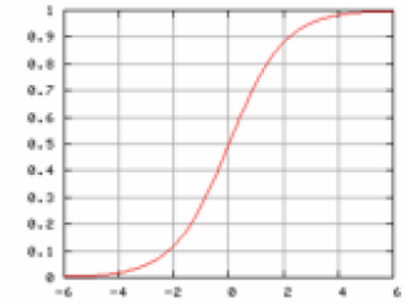
Taking $m \to \infty$, we can approximate “any function” with “any precision.”

η can be sigmoid or ReLU.

Activation functions:

ReLU: $\eta(u) = \max\{u, 0\}$ **Sigmoid:** $\eta(u) = \frac{1}{1+\exp(-u)}$





2015	Sonoda + Murata	**Unbounded**, admissible	$L_1(R^n), L_2(R^n)$

K is any compact set.

Ref：園田, “ニューラルネットワークの 積分表現理論”, 2015.

Adaptivity of deep learning

“Adaptivity”

- Deep learning shows good performances in various tasks.

  → “Adaptivity” of deep learning

  - ➢ Besov space and its variants.
  - ➢ Deep learning can outperform non-adaptive method and linear estimators.
  - ➢ Extension of the theory to more general space.

- ■ Suzuki: Adaptivity of deep ReLU network for learning in Besov and mixed smooth Besov spaces: optimal rate and curse of dimensionality. ICLR2019.
- ■ Oono&Suzuki: Approximation and Non-parametric Estimation of ResNet-type Convolutional Neural Networks. ICML2019.
- ■ Hayakawa&Suzuki: On the minimax optimality and superiority of deep neural network learning over sparse parameter spaces. arXiv:1905.09195.
- ■ Suzuki&Nitanda: Deep learning is adaptive to intrinsic dimensionality of model smoothness in anisotropic Besov space. arXiv:1910.12799, 2019.

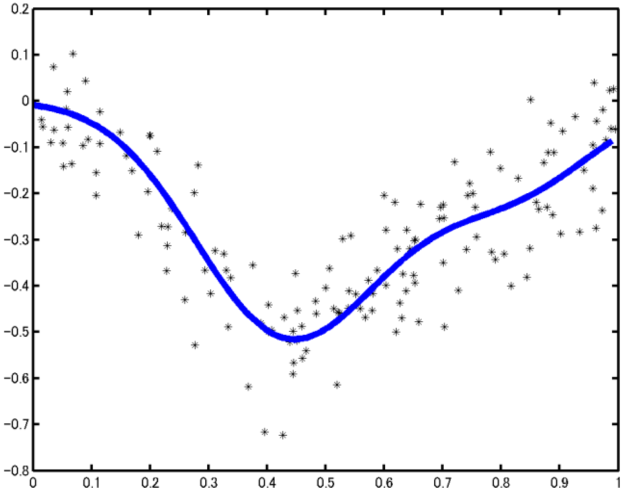
Non-parametric regression

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 $x_i \in [0,1]^d \sim P_X(X)$ (i.i.d.).

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

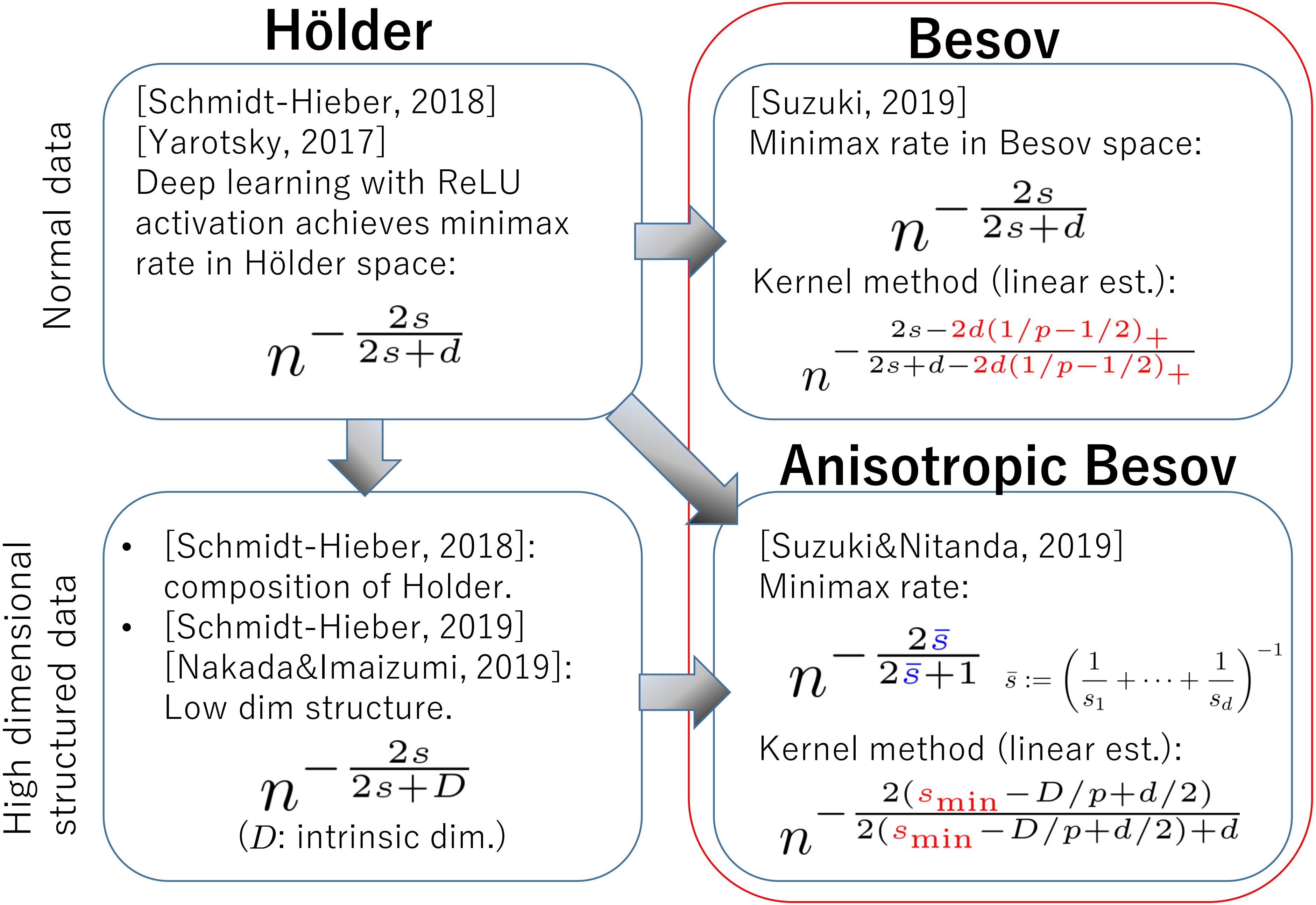


Estimation error:

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

A similar argument can be applied to classification.

Relation to existing work



Normal data

Hölder

[Schmidt-Hieber, 2018]
[Yarotsky, 2017]
Deep learning with ReLU activation achieves minimax rate in Hölder space:

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

Besov

[Suzuki, 2019]
Minimax rate in Besov space:

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

Kernel method (linear est.):

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

High dimensional structured data

- [Schmidt-Hieber, 2018]: composition of Holder.
- [Schmidt-Hieber, 2019] [Nakada&Imaizumi, 2019]: Low dim structure.

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

(D: intrinsic dim.)

Anisotropic Besov

[Suzuki&Nitanda, 2019]
Minimax rate:

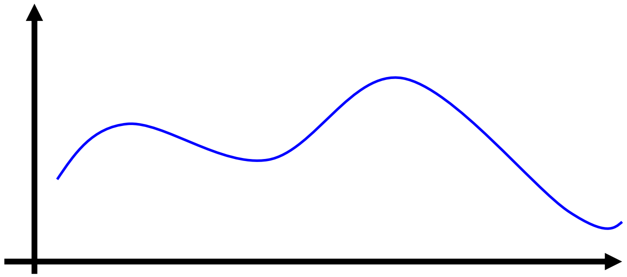
$$n^{-\frac{2\bar{s}}{2\bar{s}+1}} \quad \bar{s} := \left(\frac{1}{s_1}+\cdots+\frac{1}{s_d}\right)^{-1}$$

Kernel method (linear est.):

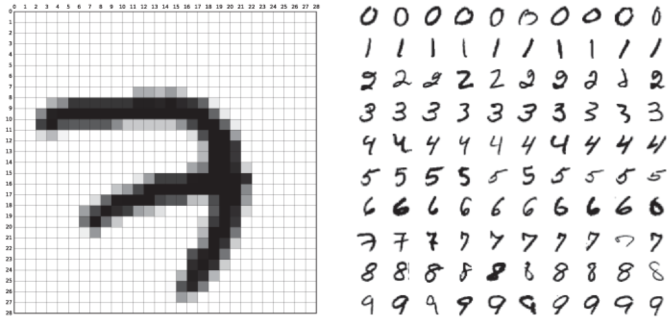
$$n^{-\frac{2(s_{\min}-D/p+d/2)}{2(s_{\min}-D/p+d/2)+d}}$$

Two quantities

- Smoothness



- Dimensionality



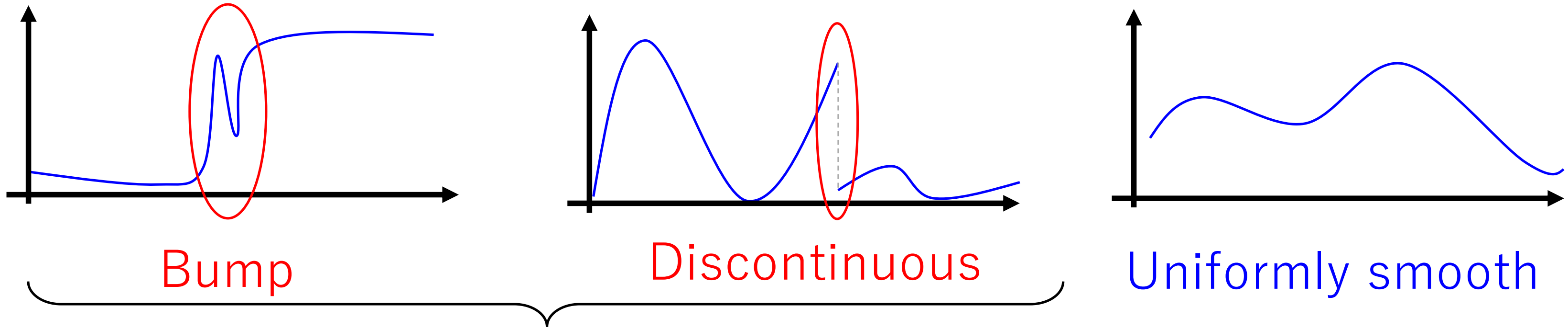
(**a**) MNIST sample belonging to the digit '7'.

(**b**) 100 samples from the MNIST training set.

Smoothness

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

In machine learning, there appears various types of functions:



Difficult **Easy**

If we overly adapt to bump, the model becomes unnecessarily large. → overfitting.
If we adapt to smooth part, bump can not be estimated. → underfitting.

“Adaptivity” is important

Theorem

Deep learning can achieve the *minimax optimal rate* to estimate functions in the Besov space ($B^s_{p,q}$).

(DL can adaptively estimate various types of functions.)

Convergence rate comparison (smoothness)

Linear estimator (shallow method)

e.g., kernel ridge regression:

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

Deep learning

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

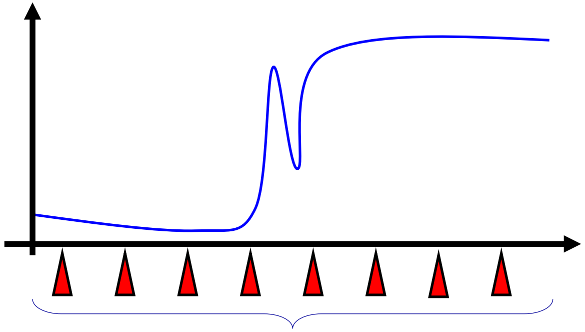
Sub-optimal

Optimal

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

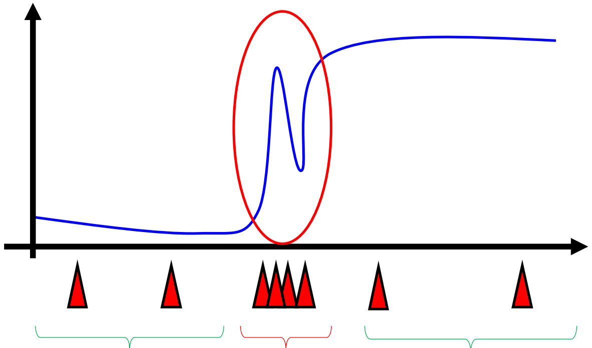
Linear method

(e.g., kernel method)



Uniform granularity

Deep learning



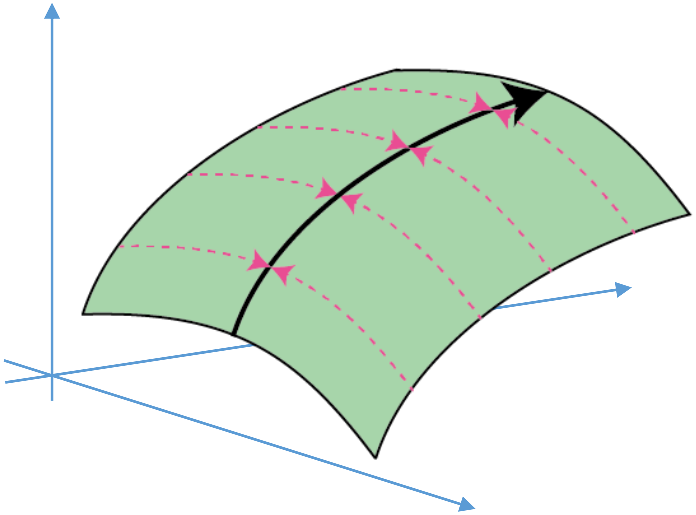
coarse fine coarse

Dimensionality

- High dimensional data
  → Curse of dimensionality

Low dimensionality of the true function:

- The true function can be very smooth (constant) in several directions.
- Data is usually distributed on a low-dimensional sub-manifold.



The estimator should find in which direction the true function is smooth.

Theorem

Deep learning is minimax-optimal also in the anisotropic Besov space.

Convergence rate comparison (dimensionality)

Linear estimator (shallow method)

Deep learning

$$n^{-\frac{2(s_{\min}-D/p+d/2)}{2(s_{\min}-D/p+d/2)+d}} \gg n^{-\frac{2\tilde{s}}{2\tilde{s}+1}}$$

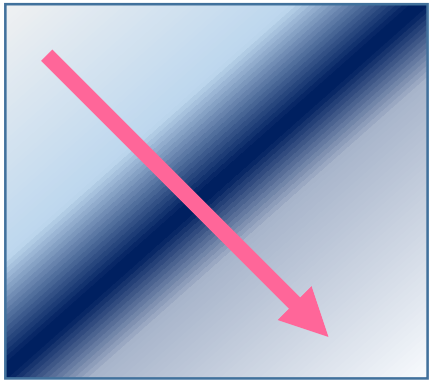
Sub-optimal

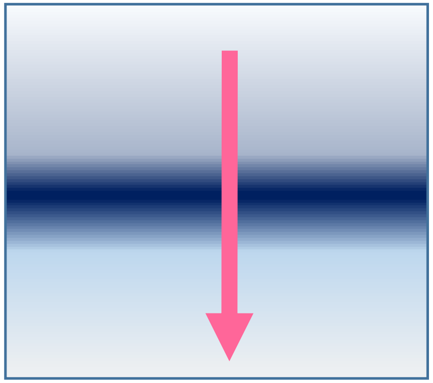
Optimal

(n: sample size, s: smoothness)

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

Linear estimator can not find smooth directions.
(lack of feature extraction ability)





Hölder, Sobolev, Besov space

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

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

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

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

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

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

Spatial homogeneity of smoothness

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

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

Smoothness

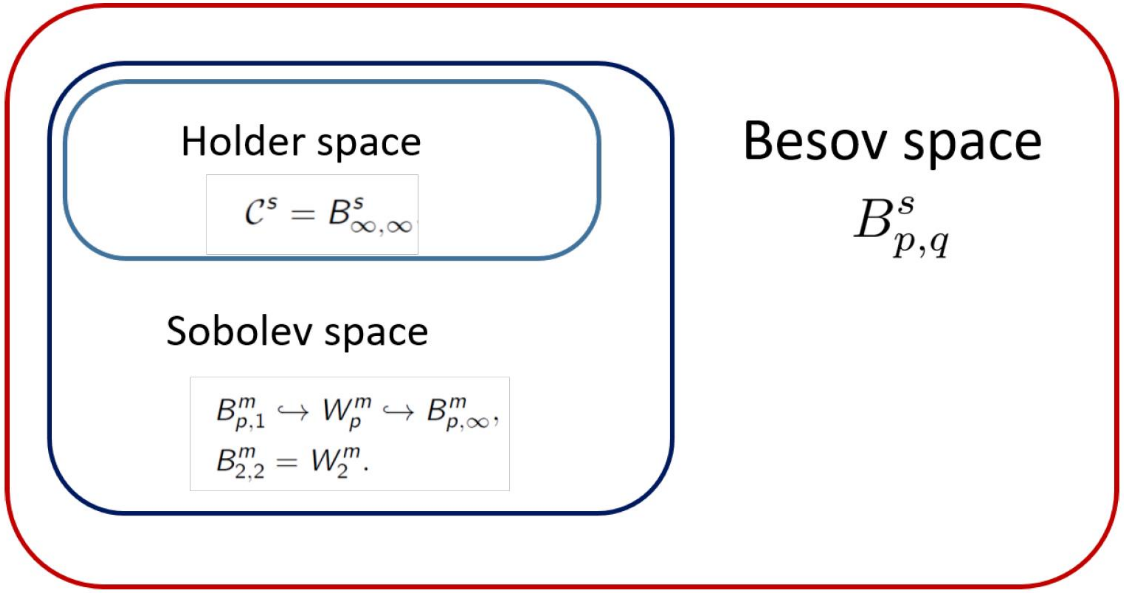
Relation between the spaces

- For $m \in \mathbb{N}$,

$$B^m_{p,1} \hookrightarrow W^m_p \hookrightarrow B^m_{p,\infty},$$
$$B^m_{2,2} = W^m_2.$$

- For $0 < s < \infty$ and $s \notin \mathbb{N}$,

$$\mathcal{C}^s = B^s_{\infty,\infty}.$$



Holder space

$\mathcal{C}^s = B^s_{\infty,\infty}$

Sobolev space

$B^m_{p,1} \hookrightarrow W^m_p \hookrightarrow B^m_{p,\infty},$
$B^m_{2,2} = W^m_2.$

Besov space

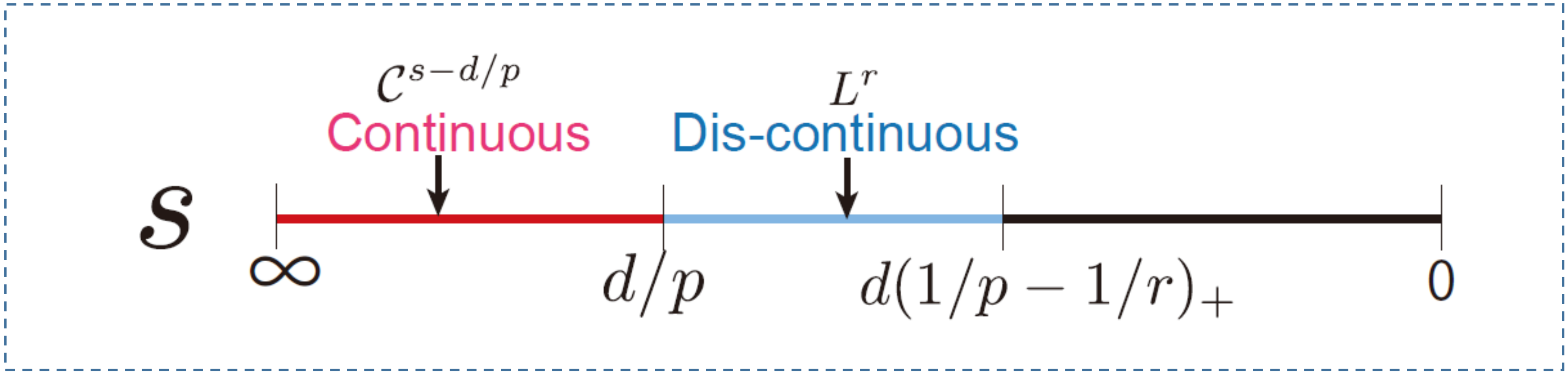
$B^s_{p,q}$

- Continuous regime: $s > d/p$

$$B^s_{p,q} \hookrightarrow C^0$$

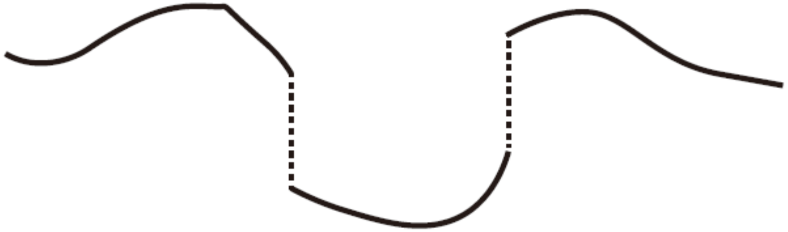
- L^r-integrability： $s \geq d(1/p - 1/r)_+$

$$B^s_{p,q} \hookrightarrow L^r$$

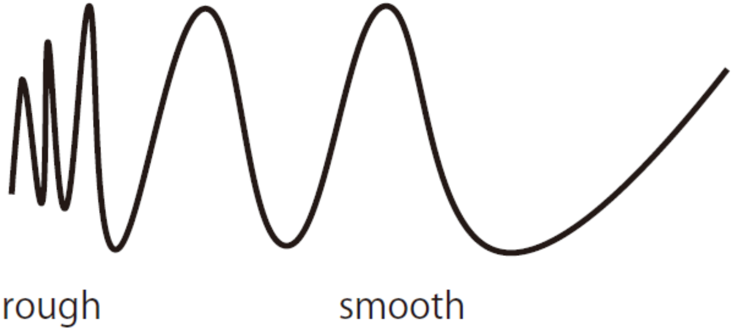


- $B^1_{1,1}([0,1]) \subset \{\text{bounded total variation}\} \subset B^1_{1,\infty}([0,1])$

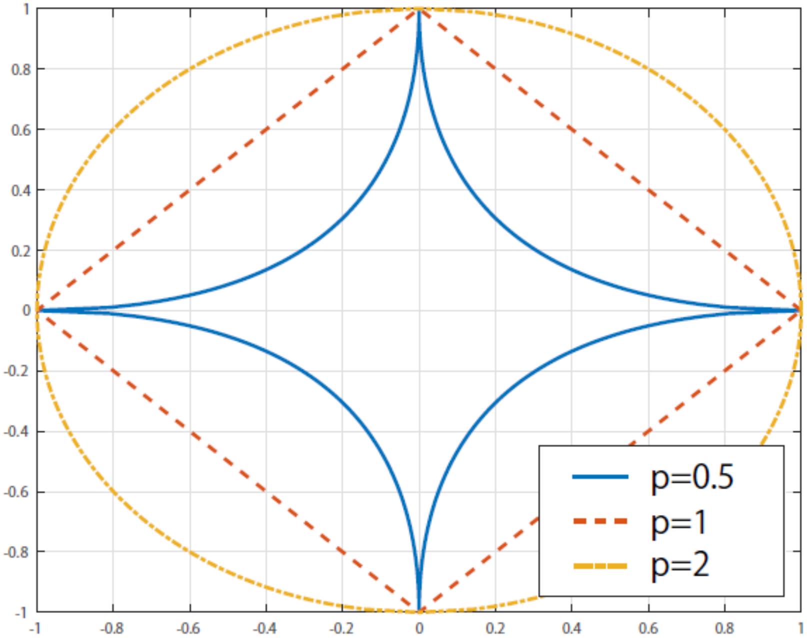
- **Discontinuity:** $d/p > s$

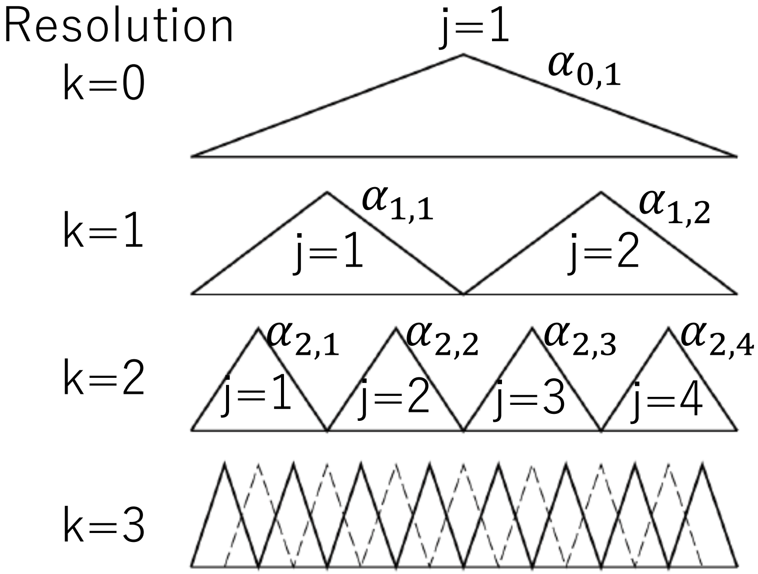


- **Spatial inhomogeneity of smoothness: small** p



Connection to wavelet





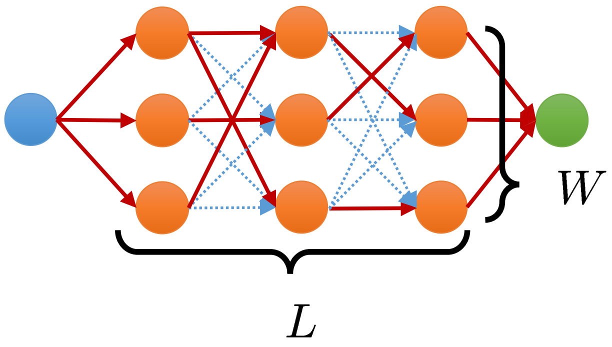
Multiresolution expansion

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

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

Sparse coefficients → spatial inhomogeneity of smoothness (non-convexity)

Deep learning model



$$f(x) = (W^{(L)}\eta(\cdot) + b^{(L)}) \circ (W^{(L-1)}\eta(\cdot) + b^{(L-1)}) \circ \cdots \circ (W^{(1)}x + b^{(1)})$$

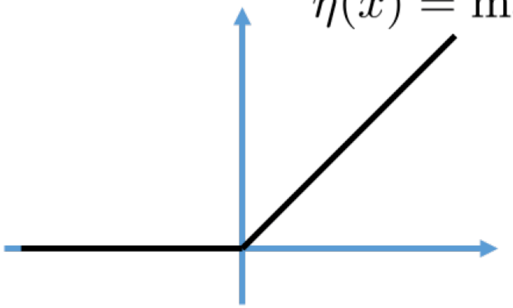
$\mathcal{F}(L, W, S, B)$

- Depth：L
- Width：W
- Sparsity：S
- Norm bound：B

Set of deep NN models

- Activation function is ReLU

$\eta(x) = \max\{x, 0\}$



Approximation in Besov space

- Assume $0 < p, q, r \leq \infty,\ 0 < s < \infty$, and following condition:

$$s > d(1/p - 1/r)_+ \qquad (L^r\text{-integrable})$$

- m is an integer s.t. $s < \min\{m, m-1+1/p\}$.

Approximation ability of deep neural network

For an integer N, let depth L, width W, sparsity S, norm bound B be

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

Then, deep NN can approximate elements in Besov space as

$$\sup_{f^\circ \in U(B^s_{p,q}([0,1]^d))} \inf_{\check{f} \in \mathcal{F}(L,W,S,B)} \|f^\circ - \check{f}\|_{L^r([0,1]^d)} \lesssim N^{-s/d}.$$

Pinkus (1999), Mhaskar (1996): $p = r$ and $1 \leq p$, ReLU activation is excluded.
Petrushev (1998): $p = r = 2$, ReLU is excluded $(s \leq k + 1 + (d-1)/2)$.

Comparison

Under the condition $s > d(1/p - 1/r)_+$, we have

$$\sup_{f^\circ \in U(B^s_{p,q}([0,1]^d))} \inf_{\check f \in \mathcal{F}(L,W,S,B)} \|f^\circ - \check f\|_{L^r([0,1]^d)} \lesssim N^{-s/d}.$$

- For $p = q = \infty$, it is reduced to Yarotsky (2016) (Hölder space)
- **Adaptive nonlinear** approx. must be used (Dung, 2011)

Linear approx. (Linear width)：

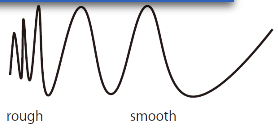
$$\begin{cases} N^{-s/d+\boxed{(1/p-1/r)_+}} & \begin{cases}\text{either} & (0<p\le r\le 2),\\ \text{or} & (2\le p\le r\le\infty),\\ \text{or} & (0<r\le p\le\infty),\end{cases} \\ N^{-s/d+\boxed{1/p-1/2}} & (0 < \end{cases}$$

$p \neq r$
is important

- Adaptivity of deep NN
- Good feature extractor

Non-adaptive approx. (N-term

$$\begin{cases} N^{-s/d+\boxed{(1/p-1/r)_+}} & (1 \\ N^{-s/d+\boxed{1/p-1/2}} & (1 < p < 2 < r \le \infty,\ s > d/p), \\ N^{-s/d} & (2 \le p < r \le \infty,\ s > d/2), \end{cases}$$



Small p

This difference does not appear for Hölder space

Related work

- Chui et al. (1994) and Bölcskei et al. (2017) dealt with a "smooth" activation with $\lim_{x \to \infty} \eta(x)/x^k \to 1, \quad \lim_{x \to -\infty} \eta(x)/x^k = 0$ with $k \geq 2$ under $1 \leq p$. Mhaskar and Micchelli (1992) studied $s = k + 1$. Mhaskar (1993) studied $k \geq 2$ and $s = k + 1$, Mhaskar (1996) considered the Sobolev space W_p^m with a "bump" activation function (excluding ReLU).

Estimation error analysis

- Least squares estimator

$$\hat{f} = \mathop{\arg\min}_{\bar{f}: f \in \mathcal{F}(L,W,S,B)} \sum_{i=1}^{n} (y_i - \bar{f}(x_i))^2$$

where $\bar{f} = \min\{\max\{f, -F\}, F\}$(clipping).

Theorem (estimation error)

Suppose $\|f^{\mathrm{o}}\|_{B^s_{p,q}} \leq 1$, $\|f^{\mathrm{o}}\|_{\infty} \leq 1$ and $0 < p, q \leq \infty,\ s > d(1/p - 1/2)_+$.

Then, by setting $N \asymp n^{\frac{d}{2s+d}}$, we have

$$\mathrm{E}[\|f^{\mathrm{o}} - \hat{f}\|^2_{L^2(P_X)}] \leq n^{-\frac{2s}{2s+d}} \log(n)^3.$$

For $p = q = \infty$, it is reduced to Schmidt-Hieber (2017).

Linear estimator

Linear estimator: an estimator which is linear to $(y_i)_{i=1}^n$.

"Shallow" method

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

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

Examples

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

Kernel ridge regression:

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

Comparison to other methods

- Linear estimators (Donoho & Johnstone, 1994)

(Kernel ridge estimator, Sieve estimator, Nadaraya-Watson, ...)

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

- Deep learning $\bigvee$

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

There appears difference when $p < 2$

When p is small (p<2), deep learning dominates
→ Spatial inhomogeneity of smoothness
(adaptivity to produce appropriate bases)

c.f., piece-wise smooth function: Imaizumi&Fukumizu, 2018.

Intuition

$$\check{f}(x) = \sum_{j=1}^{N} \underbrace{\beta_j}_{\text{Coefficient}} \underbrace{\varphi_j}_{\text{Basis}}(x)$$

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

Pre-specified: Non-adaptive method

- ➢ Kernel ridge regression, ….

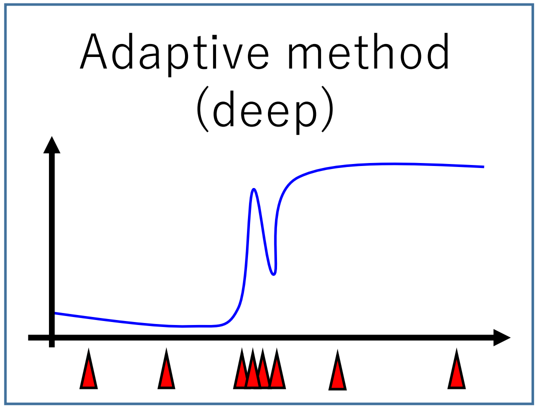
Estimated: Adaptive method

- ➢ Deep learning, sparse estimator, ….

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

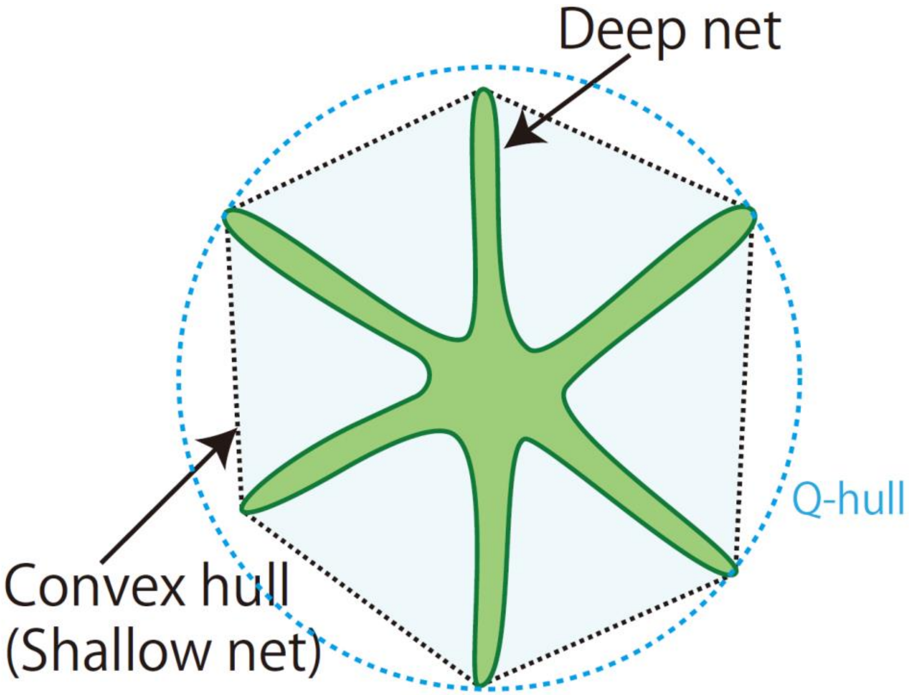
Difference between deep and sparse learning:

- Sparse:
  Choose important bases from a pre-specified set of bases.
- Deep:
  Construct bases directly.



Adaptive method (deep)

Why does this difference happen?



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

With additional conditions, it can be extended to "Q-hull."

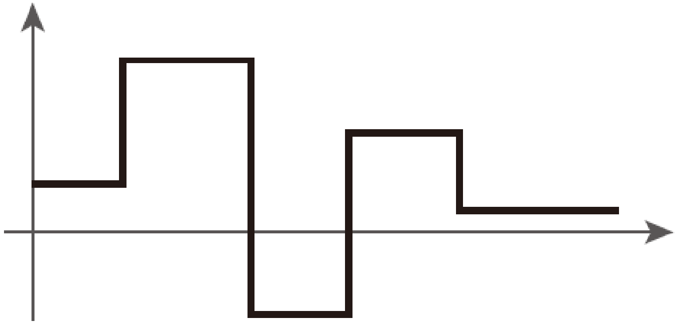
[Hayakawa&Suzuki: 2019][Donoho & Johnstone, 1994]

Simple example

[Hayakawa&Suzuki: 2019]

$$J_K = \left\{ a_0 + \sum_{i=1}^{K} a_i \mathbf{1}_{[t_i,1]} \mid t_i \in (0,1], |a_0|, \sum_{i=1}^{K} |a_i| \le 1 \right\}$$

→ Its convex hull includes the **functions of bounded variation.**



Theorem

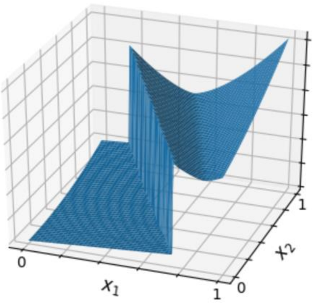
$$\inf_{\hat{f}:\text{Linear}} \sup_{f^o \in J_K} \mathrm{E}\left[\|\hat{f} - f^o\|_{L_2(P)}^2\right] \ge \Omega\left(\frac{1}{\sqrt{n}}\right).$$

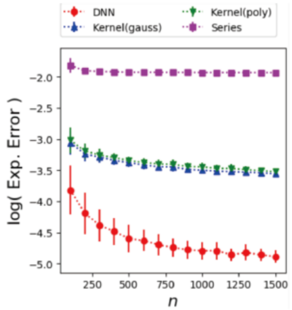
Deep learning : $o\left(\frac{1}{n}\right)$

Examples (1)

- **Piece-wise smooth function** (Imaizumi & Fukumizu, 2018)

$$f^{\circ}(x) = \sum_{k=1}^{K} \mathbf{1}_{R_k}(x) h_k(x)$$





where R_k is a region with smooth boundary and h_k is a smooth function.

➢ Deep is better than a kernel method (linear estimator).

- **Low dimensional feature extractor** (Schmidt-Hieber, 2018)

$$f^{\circ}(x) = g(w^{\top} x)$$

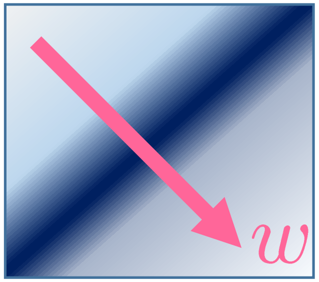
Dim. reduction

g is a univariate smooth function.

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

Deep

Wavelet series estimator
: suffers from curse of dim.



Example (2)

- Reduced rank regression

$$Y_i = UVX_i + \xi_i \quad (i = 1, \dots, n)$$

where $Y_i \in \mathbb{R}^M, X_i \in \mathbb{R}^N$ and $U \in \mathbb{R}^{M \times r}, V \in \mathbb{R}^{r \times N}$ $(r \ll M, N)$.

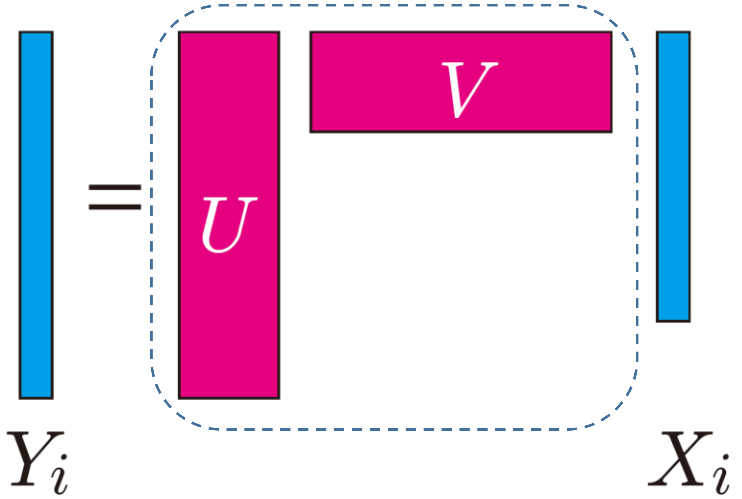
- Linear estimator $\hat{f}(x) = \sum_{i=1}^n Y_i \varphi(X_1, \dots, X_n, x)$,
- Deep learning $\hat{f}(x) = \hat{U}\hat{V}x$.

Comparison of accuracy

$$\frac{r(M+N)}{n} \ll \frac{MN}{n}$$

Deep — Shallow (LS, Ridge reg)

Low rank: non-convex



Convex hull of the low rank model is full-rank.

Curse of dimensionality

Curse of dimensionality

Estimation error bound：

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

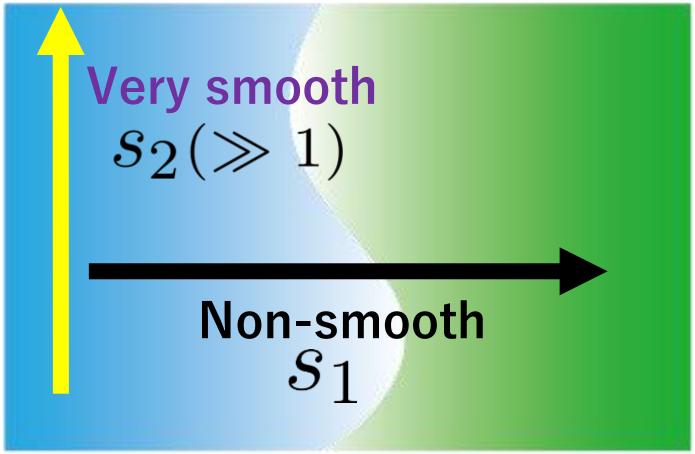
Approximation error bound :

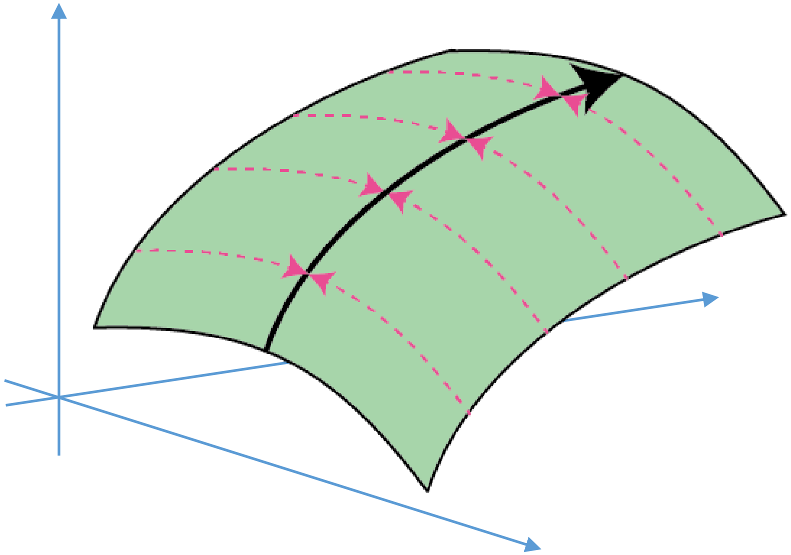
$$N^{-\frac{s}{d}}$$

→ **Curse of dimensionality**

Anisotropic Besov space

[Suzuki&Nitanda: Deep learning is adaptive to intrinsic dimensionality of model smoothness in anisotropic Besov space. arXiv:1910.12799, 2019.]





$$f^\circ \in B_{p,q}^{(s_1,\dots,s_d)} \qquad \bar{s} := \left(\frac{1}{s_1} + \cdots + \frac{1}{s_d}\right)^{-1}$$

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

Deep

- Curse of dimensionality is avoided.
- Minimax optimal.

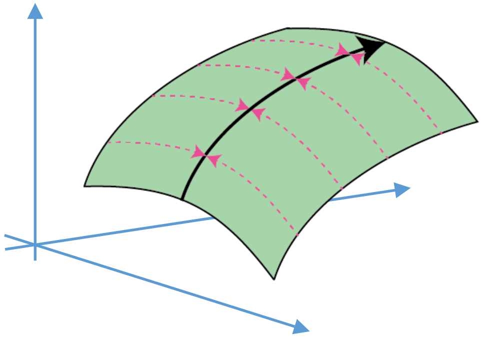
  [Ibragimov & Khas’minskii (1984), Nyssbaum (1983, 1987), Kerkyacharian et al. (2001)]

Deep composition model

$$f^\circ(x) = h_H \circ \cdots \circ h_1(x)$$

$h_\ell : \mathbb{R}^{m_\ell} \to \mathbb{R}^{m_{\ell+1}}$: included in an anisotropic Besov space ($B^{\beta^{(\ell)}}_{p,q}$).

Example:



$$f^\circ(x) = h \circ \underline{\varphi(x)}$$

Coordinate in the manifold
(feature extractor)

Theorem

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

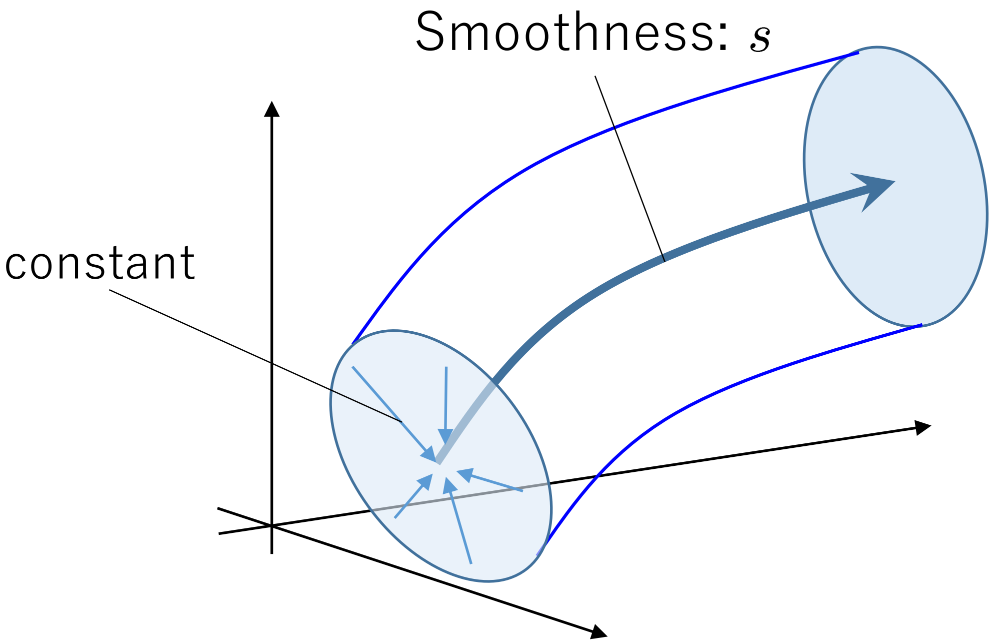
Deep learning

This is minimax optimal.

$$\tilde{\beta}^{(\ell)} := \left(\frac{1}{\beta_1^{(\ell)}} + \cdots + \frac{1}{\beta_{m_\ell}^{(\ell)}}\right)^{-1} \qquad \tilde{\beta}^{*(\ell)} := \tilde{\beta}^{(\ell)} \prod_{k=\ell+1}^{H} [(\min_j \beta_j^{(\ell)} - 1/p) \wedge 1]$$

Example

Data on smooth manifold



Intrinsic dimensionality: $d = 1$

- The true function varies only one direction in the manifold.
- Invariant against noise injection to other directions.

Deep

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

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

c.f., Manifold regression:

- Classic method: Yang & Dunson (2016), Bickel & Li (2007), Yang & Tokdar (2015)
- Deep learning: Nakada & Imaizumi (2019), Schmidt-Hieber (2019)

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

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

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

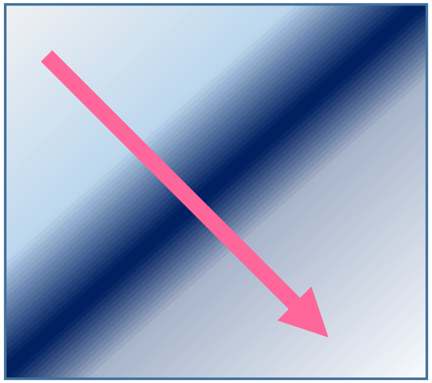
$\ll$

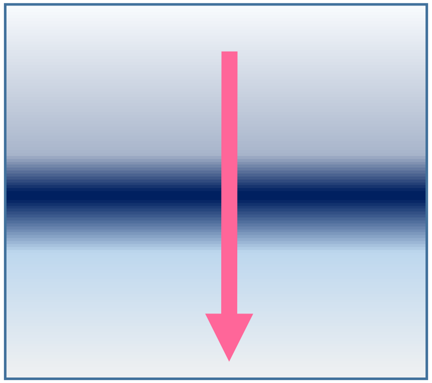
Linear estimator

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

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

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





Summary

Adaptivity of deep learning

- The ReLU-DNN has high adaptivity to shape of the target functions (spatial inhomogeneity of smoothness).

$$\|\hat{f} - f^\circ\|_{L^2(P)}^2 = O(n^{-2s/(2s+d)} \log(n)^3)$$

- DNN outperforms non-adaptive methods.

[Besov]

$$\text{(DNN)} \quad n^{-\frac{2s}{2s+d}} \ll n^{-\frac{2(s-d(1/p-1/2))}{2s+d-2d(1/p-1/2)}} \quad \text{(linear method)}$$

[Anisotropic Besov]

$$\text{(DNN)} \quad n^{-\frac{2\bar{s}}{2\bar{s}+1}} \ll n^{-\frac{2s_{\min}}{2s_{\min}+d}} \quad \text{(linear method)}$$

Deep learning $\simeq$ Sparse estimation in infinite dim. space