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
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.”

$x \xrightarrow{W_1 x} h_1(W_1x) \xrightarrow{W_2h_1(W_1x)} h_2(W_2h_1(W_1x))$

- $h_1(u) = [h_{11}(u_1), h_{12}(u_2), \ldots, h_{1d}(u_d)]^T$
- $\star$ ReLU (Rectified Linear Unit) :
- Sigmoid function :

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

$h(u) = \frac{1}{1 + e^{-u}}$
Fully connected layer

- \( \ell \)-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.”

\( \eta \) can be sigmoid or ReLU.

**Activation functions:**

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

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

Adaptivity of deep learning
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.

Non-parametric regression

\[ y_i = f^\circ(x_i) + \xi_i \quad (i = 1, \ldots, n) \]

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

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

A similar argument can be applied to classification.

Estimation error:

\[ \mathbb{E}[\|\hat{f} - f^\circ\|^2_{L^2(P)}] < ? \]
**Relation to existing work**

**Hölder**
- [Schmidt-Hieber, 2018]
- [Yarotsky, 2017]
  Deep learning with ReLU activation achieves minimax rate in Hölder space:
  \[ n \geq \frac{2s}{2s+d} \]

**Besov**
- [Yarotsky, 2017]
- [Suzuki, 2019]
  Minimax rate in Besov space:
  \[ n \geq \frac{2s}{2s+d} \]
  Kernel method (linear est.):
  \[ n \geq \frac{2s - 2d(1/p - 1/2)_+}{2s + d - 2d(1/p - 1/2)_+} \]

**Anisotropic Besov**
- [Schmidt-Hieber, 2018]: composition of Hölder.
- [Schmidt-Hieber, 2019]
- [Nakada & Imaizumi, 2019]: Low dim structure.

- [Suzuki & Nitanda, 2019]
  Minimax rate:
  \[ n \geq \frac{2\bar{s}}{2\bar{s} + 1} \]
  \( \bar{s} := \left( \frac{1}{s_1} + \cdots + \frac{1}{s_d} \right)^{-1} \)
  Kernel method (linear est.):
  \[ n \geq \frac{2(s_{\min} - D/p + d/2)}{2(s_{\min} - D/p + d/2) + d} \]
Two quantities

- Smoothness

- Dimensionality

(a) MNIST sample belonging to the digit ‘7’. (b) 100 samples from the MNIST training set.
Smoothness


In machine learning, there appears various types of functions:

- **Bump**
- **Discontinuous**
- **Uniformly smooth**

**Difficult**

If we overly adapt to bump, the model becomes unnecessarily large. $\rightarrow$ overfitting.

If we adapt to smooth part, bump can not be estimated. $\rightarrow$ underfitting.

"Adaptivity" is important

---

**Theorem**

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

(DL can adaptively estimate various types of functions.)
Convergence rate comparison (smoothness)

Linear estimator (shallow method)

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

\[ n \leq \frac{2s - 2(1/p - 1/2)_+}{2s + 1 - 2(1/p - 1/2)_+} \]

Sub-optimal

Deep learning

\[ n \geq \frac{2s}{2s + 1} \]

Optimal

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

Linear method (e.g., kernel method)

Deep learning
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

\[
\hat{n} \sim \frac{2(s_{\text{min}} - D/p + d/2)}{2(s_{\text{min}} - D/p + d/2) + d} \quad \gg \quad \hat{n} \sim \frac{2\hat{s}}{2\hat{s} + 1}
\]

Sub-optimal  
Optimal  
\[\hat{s} := \left(\frac{1}{s_1} + \cdots + \frac{1}{s_d}\right)^{-1}\]

*(\hat{n}: sample size, \(s\): smoothness)*

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** \( (C^\beta(\Omega)) \)

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

- **Sobolev space** \( (W^k_p(\Omega)) \)

\[ \| f \|_{W^k_p} = \left( \sum_{|\alpha| \leq k} \| D^\alpha f \|_{L^p(\Omega)}^p \right)^{1/p} \]

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

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

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

Spatial homogeneity of smoothness

Smoothness
Relation between the spaces

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

\[
B_{p,1}^m \leftrightarrow W_p^m \leftrightarrow B_{p,\infty}^m,
\]

\[
B_{2,2}^m = W_2^m.
\]

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

\[
C^s = B_{\infty,\infty}^s.
\]
• Continuous regime: \( s > d/p \)

\[
B_{p,q}^s \hookrightarrow C^0
\]

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

\[
B_{p,q}^s \hookrightarrow L^r
\]

- \( B_{1,1}^1([0,1]) \subset \{ \text{bounded total variation} \} \subset B_{1,\infty}^1([0,1]) \)
• Discontinuity: $d/p > s$

• Spatial inhomogeneity of smoothness: small $p$

![Diagram showing a transition from rough to smooth with a notation $d/p > s$]
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: \( B \)

- 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:
  \[ s > d(1/p - 1/r)_+ \]  
  \( (L^r \text{-integrable}) \)

- $m$ 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

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

Then, deep NN can approximate elements in Besov space as

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

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

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

- For $p = q = \infty$, it is reduced to Yarotsky (2016) (Hölder space)

- **Adaptive nonlinear** approx. must be used (Dung, 2011)

**Linear approx. (Linear width):**

\[
\begin{cases}
N^{-s/d + \frac{1}{p-1/r} +} \\
N^{-s/d + \frac{1}{p-1/2}}
\end{cases}
\begin{aligned}
&
\left\{ \begin{array}{ll}
& \text{either} \quad (0 < p \leq r \leq 2), \\
& \text{or} \quad (2 \leq p \leq r \leq \infty), \\
& \text{or} \quad (0 < r \leq p \leq \infty),
\end{array} \right.
\end{aligned}
\]

**Non-adaptive approx. (N-term approx., Kolmogorov width):**

\[
\begin{cases}
N^{-s/d + \frac{1}{p-1/r} +} \\
N^{-s/d + \frac{1}{p-1/2}} \\
N^{-s/d}
\end{cases}
\begin{aligned}
&
\left\{ \begin{array}{ll}
& (1 < p < 2 < r \leq \infty, \; s > d/p), \\
& (2 \leq p < r \leq \infty, \; s > d/2),
\end{array} \right.
\end{aligned}
\]

- **Adaptivity of deep NN**
- **Good feature extractor**

This difference does not appear for Hölder space
Chui et al. (1994) and Bölcskei et al. (2017) dealt with a “smooth” activation with $\lim_{x \to \infty} \eta(x)/x^k \to 1$, $\lim_{x \to -\infty} \eta(x)/x^k = 0$ 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 $W^m_p$ with a “bump” activation function (excluding ReLU).
Estimation error analysis

- Least squares estimator

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

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

**Theorem (estimation error)**

Suppose \( \|f^0\|_{B_{p,q}^s} \leq 1, \|f^0\|_{\infty} \leq 1 \) and \( 0 < p, q \leq \infty, s > d(1/p - 1/2)_+ \). Then, by setting \( N \asymp n^{\frac{d}{2s+d}} \), we have

\[ \mathbb{E}[\|f^0 - \hat{f}\|_{L^2(P_X)}^2] \leq n^{-\frac{2s}{2s+d}} \log(n)^3. \]

For \( p = q = \infty \), it is reduced to Schmidt-Hieber (2017).
Linear estimator: an estimator which is linear to $(y_i)_{i=1}^n$.

“Shallow” method

\[ X_n = (x_1, \ldots, x_n) \]

\[ \hat{f}(x) = \sum_{i=1}^{n} \varphi(x; X_n)y_i \]

Examples

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

Kernel ridge regression:

\[ \hat{f}(x) = K_{x,X}(K_{X,X} + \lambda I)^{-1}Y \]
Comparison to other methods

• Linear estimators  
  (Donoho & Johnstone, 1994)  
  (Kernel ridge estimator, Sieve estimator, Nadaraya-Watson, ...)

\[ n \leq \frac{2s - 2d(1/p - 1/2)}{2s + d - 2d(1/p - 1/2)} \]

• Deep learning

\[ n \leq \frac{2s}{2s + d} \]

There appears difference when \( p < 2 \)

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

Difference between deep and sparse learning:

- **Sparse:** Choose important bases from a pre-specified set of bases.
- **Deep:** Construct bases directly.
Why does this difference happen?

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

Simple example

[Hayakawa & Suzuki: 2019]

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

→ Its convex hull includes the functions of bounded variation.

\[ \inf_{\hat{f} : \text{Linear}} \sup_{f^0 \in J_K} \mathbb{E} \left[ \| \hat{f} - f^0 \|_{L_2(P)}^2 \right] \geq \Omega \left( \frac{1}{\sqrt{n}} \right). \]

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

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

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

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)

\[
f^\circ(x) = g(w^\top x)
\]

\( g \) is a univariate smooth function.

\[
n \frac{2s}{2s+1} \ll n \frac{2s}{2s+d}
\]

Deep \hspace{1cm} Wavelet series estimator

: suffers from curse of dim.
Example (2)

• Reduced rank regression

\[ Y_i = U V X_i + \xi_i \quad (i = 1, \ldots, n) \]

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

Comparison of accuracy

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

Deep (LS, Ridge reg)

Shallow

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

Estimation error bound:

\[ n \sim \frac{2s}{2s+d} \]

Approximation error bound:

\[ N \sim \frac{s}{d} \]

→ Curse of dimensionality
Anisotropic Besov space


\[ f^\circ \in B_p^{(s_1, \ldots, s_d)} \quad \bar{s} := \left( \frac{1}{s_1} + \cdots + \frac{1}{s_d} \right)^{-1} \]

\[ \mathcal{N} \sim \frac{2\bar{s}}{2\bar{s}+1} \]

- Curse of dimensionality is avoided.
- Minimax optimal.

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

\[ f^\circ (x) = h_H \circ \cdots \circ h_1(x) \]

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

Example:

\[ f^\circ (x) = h \circ \varphi(x) \]

Coordinate in the manifold (feature extractor)

**Theorem**

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

Deep learning

\[ \tilde{\beta}(\ell) := \left( \frac{1}{\beta_1(\ell)} + \cdots + \frac{1}{\beta_{m_\ell}(\ell)} \right)^{-1} \]

\[ \beta^*(\ell) := \tilde{\beta}(\ell) \prod_{k=\ell+1}^{H} \left( \min_j \beta_j^{(\ell)} - 1/p \right) \land 1 \]

This is minimax optimal.
Data on smooth manifold

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

Intrinsic dimensionality: \( d = 1 \)

Naïve evaluation: \( n \frac{2s}{2s+d} \)

c.f., Manifold regression:
Comparison to linear estimator

\[ f^\circ(x) = g(Wx) \quad (W \in \mathbb{R}^{D \times d}, \ g \in B_p^s([0, 1]^D)) \]

\[ f^\circ \] depends only \( D \)-dimensional subspace.

Deep

\[
\mathcal{O}(\frac{2s}{2s + D})
\]

(\( \mathcal{O}(\frac{2s}{2s + d/2}) \) when \( D = \frac{d}{2} \))

Linear estimator

\[
\mathcal{O}(\frac{2(s - D/p + d/2)}{2(s - D/p + d/2) + d}) \lor \mathcal{O}(\frac{2s}{2s + D})
\]

(\( \mathcal{O}(\frac{2s}{2s + d}) \) when \( D = \frac{d}{2} \) and \( p = 1 \))

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

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

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

• DNN outperforms non-adaptive methods.

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

[Anisotropic Besov] \( n^{-\frac{2s}{2\tilde{s}+1}} \ll n^{-\frac{2s_{\min}}{2s_{\min}+d}} \) (linear method)

Deep learning \( \approx \) Sparse estimation in infinite dim. space