Presentation on Bauer and Kohler (2019)[1]

Chenceng Fang

Bonn Graduate School of Economics

2022-06-27

[1] Bauer, B. and Kohler, M. (2019), On Deep Learning as a Remedy for the Curse of Dimensionality in Nonparametric Regression, The Annuals of Statistics 47(4): 2261-2285

Curse of Dimensionality

Recap: Curse of Dimensionality

- Write a **one-dimensional** non-parametric model

$$Y = g(X) + U, E[U|X] = 0$$

- Main result:

$$\int (\hat{g}(x) - g(x))^2 f_X(x) dx = O_p(K_n/n + K_n^{-2\alpha})$$

- In case of fastest rate of convergence, the variance and squared bias need to converge at the same rate. Then,

$$\int (\hat{g}(x) - g(x))^2 f_X(x) dx = \boxed{O_p(n^{-\frac{2\alpha}{1+2\alpha}})}$$

- With more smoothness, i.e., higher α, we have higher rate of convergence.

Recap: Curse of Dimensionality

- However, now write a **higher-dimensional** non-parametric model

$$Y = g(X^{(1)}, X^{(2)}, \dots, X^{(d)}) + U, E[U|X] = 0$$

- Then, the mean square rate of convergence is

$$O_p(K_n^d/n + K_n^{-2\alpha})$$

.

- By following similar arguments as before, $K_n \asymp n^{\frac{1}{d+2\alpha}}$ in case of fastest rate of convergence. Then, mean square rate of convergence becomes

$$\boxed{O_p(n^{-\frac{2\alpha}{d+2\alpha}})}$$

- Here comes the problem of **curse of dimensionality**

Main result of paper

Additive and Interaction Model

- Stone (1985) assumes an additivity condition:

$$m(X^{(1)}, X^{(2)}, \cdots, X^{(d)}) = m_1(X^{(1)}) + \cdots + m_d(X^{(d)})$$

  Now, the optimal rate of convergence of additive model is $\boxed{O_p(n^{-\frac{2p}{1+2p}})}$.

- Stone (1994) generalized this to interaction model. Suppose for some $d^* \in \{1, \ldots, d\}$, the model is:

$$m(X) = \sum_{I \subseteq \{1,\ldots,d\}, |I| = d^*} m_I(X_I)$$

  where $X = (X^{(1)}, \ldots, X^{(d)})^T \in \mathbb{R}^d$, all m_I are smooth functions defined on $\mathbb{R}^{|I|}$ and for $I = \{i_1, \cdots, i_{d^*}\}$ with $1 \leq i_1 \leq \cdots \leq i_{d^*} \leq d$, the abbreviation $X_I = (X^{(i_1)}, \cdots, X^{(i_{d^*})})^T$ is used. \ Now, the optimal convergence rate of interaction model is $\boxed{O_p(n^{-\frac{2p}{d^*+2p}})}$.

Single Index Models and Projection Pursuit

- Single index model:

$$m(X) = g(a^T X), (X \in \mathbb{R}^d)$$

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

- Single index model is extended to so called projection pursuit:

$$m(X) = \sum_{k=1}^{K} g_k(a_k^T X), (X \in \mathbb{R}^d)$$

  where $K \in \mathbb{N}$, $g_k : \mathbb{R} \to \mathbb{R}$ and $a_k \in \mathbb{R}^d$.

- Horowitz and Mammen (2007) further studies the following model (simplified):

$$m(X) = F(m_1(X^1) + \cdots + m_d(X^{(d)})) + U$$

- A univariate rates of convergence, i.e., $\boxed{O_p(n^{-\frac{2p}{1+2p}})}$ has been proved for above three models, up to some logarithmic factor.

Generalized Hierarchical Interaction Model

- Motivation: Applications in complex technical system, which are constructed in modular form. Each modulars depends only on a few of the components of a high-dimensional input.

- Let $d \in \mathbb{N}$, $d^* \in \{1, \cdots, d\}$ and $m : \mathbb{R}^d \to \mathbb{R}$

  1. m is a gerneralized hierarchical interaction model of order d^* and level 0, if $\exists \quad a_1, \cdots, a_{d^*} \in \mathbb{R}^d$ and for $f : \mathbb{R}^{d^*} \to \mathbb{R}$, s.t. for all $X \in \mathbb{R}^d$,
  $$m(X) = f(a_1^T X, \cdots, a_{d^*}^T X)$$
  2. m is a gerneralized hierarchical interaction model of order d^* and level $l + 1$, if $\exists \quad K \in \mathbb{N}$, $g_k : \mathbb{R}^{d^*} \to \mathbb{R}(k = 1, \cdots, K)$ and $f_{1,k}, \cdots, f_{d^*,k} : \mathbb{R}^d \to \mathbb{R}(k = 1, \cdots, K)$, s.t. $f_{1,k}, \cdots, f_{d^*,k}(k = 1, \cdots, K)$ satisfy a generalized hierarchical interation model of order d^* and level l,for all $X \in \mathbb{R}^d$,:
  $$m(X) = \sum_{k=1}^{K} g_k(f_{1,k}(X), \cdots, f_{d^*,k}(X))$$

Generalized Hierarchical Interaction Model (Example)

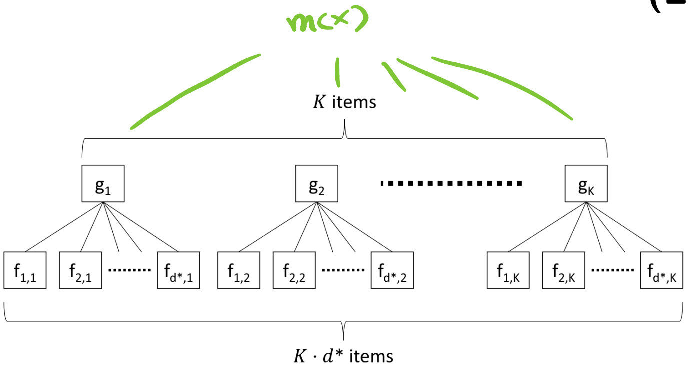


FIG. 2. *Illustration of the components of a function from* $\mathcal{H}^{(l)}$.

Multilayer Feedforward Neural Networks

- Single hidden layer neural network:
  1. Barron(1994): L_2 error has a dimensionless rate of $\boxed{n^{-1/2}}$ (up to some logarithmic factor), provided the Fourier transform has a finite first moment.
  2. McCaffrey and Gallant(1994): L_2 error has a rate of $\boxed{n^{-\frac{2p}{2p+d+5}+\epsilon}}$ for a suitably defined single hidden layer neural network estimate for (p, C)-smooth functions.
- Two and multi-layer neural network:
  1. Kohler and Krzyżak (2005): Suitable two layer nn estimates achieve a rate of convergence of $\boxed{n^{-\frac{2p}{2p+d^*}}}$ (up to some logarithmic factor) for (p, C)-smooth interaction models with $\boxed{p \le 1}$.
  2. Kohler and Krzyżak (2017): Suitable defined multilayer nn estimates achieve a rate of convergence of $\boxed{n^{-\frac{2p}{2p+d^*}}}$ (up to some logarithmic factor) for (p, C)-smooth interaction models with $\boxed{p \le 1}$.

Multilayer Feedforward Neural Networks

- Bauer and Kohler (2019): L_2 errors of least squares nn estimates achieve the rate of convergence $\boxed{n^{-\frac{2p}{2p+d^*}}}$ (up to some logarithmic factor) for (p, C)-smooth generalized hierarchical interaction model of given order d^* and given level l. Here, $\boxed{p > 0}$ might be arbitrarily large.
- Similar rates have been obtained in the literature. However, they have much more stringent assumptions on the functional class the regression function belongs to.
- To achieve the above-mentioned rate, completely new approximation results for sparse neural networks with several hidden layers were needed.

Sparse Multilayer Feedforward Neural Networks

- For $M^* \in \mathbb{N}$, $d \in \mathbb{N}$, $d^* \in \{1, \ldots, d\}$ and $\alpha > 0$, we denote the set of all functions $f : \mathbb{R}^d \to \mathbb{R}$ that satisfy:

$$f(X) = \sum_{i=1}^{M^*} \mu_i \cdot \sigma \left(\sum_{j=1}^{4d^*} \lambda_{i,j} \cdot \sigma \left(\sum_{v=1}^{d} \theta_{i,j,v} \cdot X^{(v)} + \theta_{i,j,0} \right) + \lambda_{i,0} \right) + \mu_0$$

$X \in \mathbb{R}^d$ for some μ_i, $\lambda_{i,j}$ and $\theta_{i,j,v} \in \mathbb{R}$, where $|\mu_i| \leq \alpha$, $|\lambda_{i,j}| \leq \alpha$ and $|\theta_{i,j,v}| \leq \alpha$ for all $i \in \{0, 1, \ldots, M^*\}$, $i \in \{0, \ldots, 4d^*\}$, $v \in \{0, \ldots, d\}$, by $\boxed{\mathcal{F}^{(\text{neural networks})}_{M^*,d^*,d,\alpha}}$.

- The neural network has only $W(\mathcal{F}^{(\text{neural networks})}_{M^*,d^*,d,\alpha})$ weights.

$$\begin{aligned} W(\mathcal{F}^{(\text{neural networks})}_{M^*,d^*,d,\alpha}) &= M^* + 1 + M^*(4d^* + 1) + M^* 4d^*(d + 1) \\ &= M^*(4d^*(d + 2) + 2) + 1 \end{aligned}$$

$\# n: 4d^* \cdot M^*$

Sparse Multilayer Feedforward Neural Network

$d^* = 1$

$M^* = 3$

$d = 5$

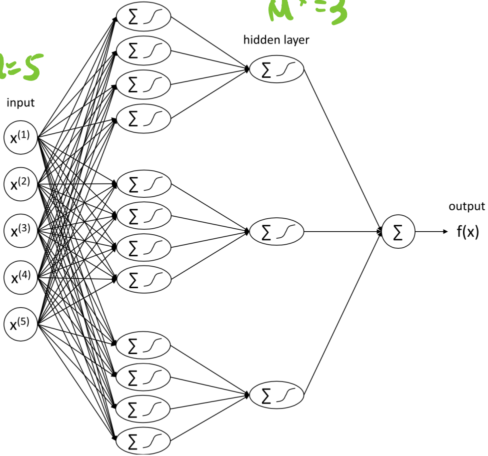


FIG. 1. *A not completely connected neural network* $f : \mathbb{R}^5 \to \mathbb{R}$ *from* $\mathcal{F}^{(\text{neural networks})}_{3,1,5,\alpha}$ *with the structure* $f(x) = \sum_{i=1}^{3} \mu_i \cdot \sigma(\sum_{j=1}^{4} \lambda_{i,j} \cdot \sigma(\sum_{v=1}^{5} \theta_{i,j,v} \cdot x^{(v)}))$ *(all weights with an index including zero neglected for a clear illustration).*

Hierarchical Neural Networks

- For $l = 0$,we define our space of hierarchical neural networks by

$$\mathcal{H}^{(0)} = \mathcal{F}^{(\text{neural networks})}_{M^*,d^*,d,\alpha}$$

- For $l > 0$, we define recursively

$$\mathcal{H}^{(l)} = \left\{ h : \mathbb{R}^d \to \mathbb{R} : h(X) = \sum_{k=1}^{K} g_k(f_{1,k}(X), \ldots, f_{d^*,k}(X)) \right\}$$

for some $g_k \in \mathcal{F}^{(\text{neural networks})}_{M^*,d^*,d^*,\alpha}$ and $f_{j,k} \in \mathcal{H}^{(l-1)}$

- Theorem: With $M^* = \lceil c_{56} \cdot n^{\frac{d^*}{2p+d^*}} \rceil$ and some other assumptions,

$$\mathbf{E} \int |m_n(x) - m(x)|^2 \mathbf{P}_X(dx) \leq \boxed{c_4 \cdot log(n)^3 \cdot n^{-\frac{2p}{2p+d^*}}}$$

holds for sufficiently large n.

Hierarchical Neural Networks (Example)

$h(x)$ $\quad$ $\mathcal{H}^{(l)}$

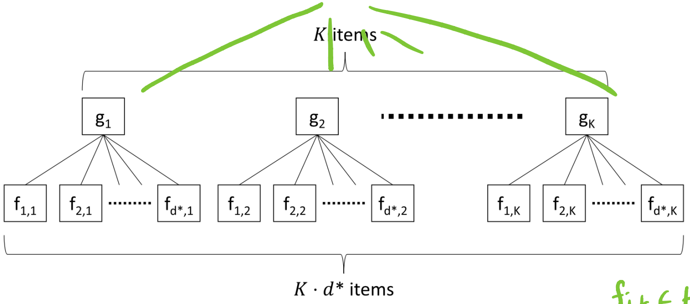


FIG. 2. *Illustration of the components of a function from* $\mathcal{H}^{(l)}$.

$f_{j,k} \in \mathcal{H}^{(0)} = \mathcal{F}^{nn}_{M^*=3,\, d^*=2,\, d=10,\, \alpha}$

$g_k \in \mathcal{F}^{nn}_{M^*=3,\, d^*=2,\, d=2,\, \alpha}$

Simulations

Settings

- Alternative approaches:
  1. Simple Nearest Neighbor Estimate (*neighbor*)
  2. Interpolation with radical basis functions (*RBF*)
  3. Fully connected neural networks with predefined numbers of layers but adaptively chosen numbers of neurons per layer (*neural-1*, *neural-3*, *neural-6*)
- Model functions:
  1. m_1 represents some ordinary general hierarchical interaction models
  2. m_4 is an additive model with $d^* = 1$
  3. m_5 is an interaction model with $d^* = d$
- Data generation:

$$Y = m_i(X) + \sigma_i \cdot \lambda_i \cdot \epsilon$$

where $i \in \{1, 2, 3, 4, 5, 6\}$ and $j \in \{1, 2\}$. X is uniformly distributed on $[0, 1]^d$ and ϵ is standard normally distributed and independent of X. σ_i and λ_i are predetermined.

Results

TABLE 1
Median and IQR of the scaled empirical L_2 error of estimates for m_1, m_2 and m_3

	m_1			
Noise	**5%**		**20%**	
Sample size	$n = 100$	$n = 200$	$n = 100$	$n = 200$
$\bar{\varepsilon}_{L_2,\bar{N}}$ ***(avg)***	**596.52**	**597.61**	**596.51**	**597.63**
Approach	**Median (IQR)**	**Median (IQR)**	**Median (IQR)**	**Median (IQR)**
neural-1	0.2622 (2.7248)	0.1064 (0.3507)	0.3004 (2.1813)	0.1709 (3.8163)
neural-3	0.1981 (0.4732)	0.0609 (0.1507)	0.2784 (0.4962)	0.0848 (0.1239
neural-6	0.2953 (0.9293)	0.1207 (0.1672)	0.2663 (0.5703	0.1106 (0.2412)
neural-x	**0.0497 (0.2838)**	**0.0376 (0.2387)**	**0.0596 (0.2460)**	**0.0200 (0.1914)**
RBF	0.3095 (0.4696)	0.1423 (0.0473)	0.3182 (0.5628)	0.1644 (0.0639)
neighbor	0.6243 (0.1529)	0.5398 (0.1469)	0.6303 (0.1014)	0.5455 (0.1562)

Results

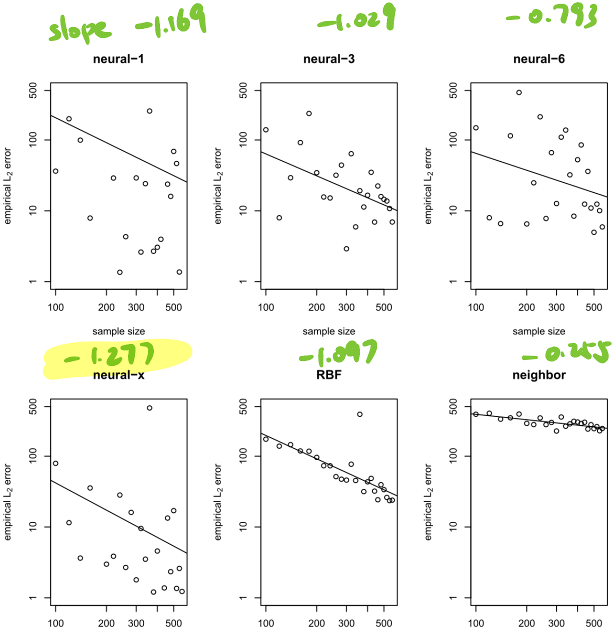
TABLE 2
Median and IQR of the scaled empirical L_2 error of estimates for m_4, m_5 and m_6

	m_4			
Noise	**5%**		**20%**	
Sample size	$n = 100$	$n = 200$	$n = 100$	$n = 200$
$\bar{\varepsilon}_{L_2,\bar{N}}$ ***(avg)***	**1.60**	**1.59**	**1.61**	**1.61**
Approach	**Median (IQR)**	**Median (IQR)**	**Median (IQR)**	**Median (IQR)**
neural-1	**0.0140** (0.0040)	**0.0050** (0.0020)	**0.0370** (0.0150)	**0.0240** (0.0090)
neural-3	0.0160 (0.0060)	0.0080 (0.0020)	0.0450 (0.0110)	**0.0240** (0.0050)
neural-6	0.0210 (0.0080)	0.0090 (0.0030)	0.0530 (0.0130)	0.0290 (0.0090)
neural-x	0.0311 (0.1026)	0.0085 (0.0205)	0.2623 (1.5689)	0.1042 (0.2296)
RBF	0.0188 (0.0084)	0.0148 (0.0030)	0.1594 (0.0589)	0.1386 (0.0299)
neighbor	0.3024 (0.07565)	0.2033 (0.0321)	0.2868 (0.0952)	0.2211 (0.0355)

Results

	m_5			
Noise	5%		20%	
Sample size	$\boldsymbol{n = 100}$	$\boldsymbol{n = 200}$	$\boldsymbol{n = 100}$	$\boldsymbol{n = 200}$
$\bar{\varepsilon}_{L_2,\bar{N}}$ ***(avg)***	**1.49**	**1.49**	**1.49**	**1.49**
Approach	**Median (IQR)**	**Median (IQR)**	**Median (IQR)**	**Median (IQR)**
neural-1	0.7246 (9.3962)	0.0648 (0.0879)	2.0865 (75.4682)	0.6659 (26.0015)
neural-3	0.3954 (0.9887)	0.1087 (0.1909)	1.5671 (7.0394)	0.2370 (1.4065)
neural-6	0.1023 (0.3572)	0.0716 (0.0760)	0.2482 (0.6611)	**0.0836 (0.1646)**
neural-x	0.1386 (0.4205)	0.0637 (0.0499)	0.3699 (1.3039)	0.1854 (0.3660)
RBF	**0.0127 (0.0044)**	**0.0112 (0.0033)**	**0.1445 (0.0671)**	0.1352 (0.0298)
neighbor	0.3263 (0.0842)	0.2471 (0.0381)	0.3360 (0.0707)	0.2620 (0.0464)

Results



slope -1.169
-1.029
-0.793
neural−1
neural−3
neural−6
-1.277
-1.097
-0.255
neural−x
RBF
neighbor
empirical L_2 error
sample size
1
10
100
500
100
200
300
500

Vielen Dank! ;) 🥳🥳🥳