Nonparametric Logistic Regression with Deep Neural Networks

Atsutomo Yara[1], Yoshikazu Terada[1, 2]

[1] Graduate School of Engineering Science, Osaka University
[2] Center for Advanced Intelligence Project (AIP), RIKEN

Sep 30th

Contents

Introduction

Classification and Estimation of Conditional Probability

- **Setting of Classification Problem**
  - Consider the K class classification problem.
    - $\mathcal{X} \subset \mathbb{R}^d$: input space, $\mathcal{Y} = \{1, \ldots, K\}$: output space.
    - P : distribution on $\mathcal{X} \times \mathcal{Y}$, $P_{\boldsymbol{X}}$: marginal distribution of X.
    - $(\boldsymbol{X}, Y), (\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n) \sim_{i.i.d.} P$
    - $\mathcal{D}_n := \{(\boldsymbol{X}_i, Y_i)\}_{i=1}^n$: observed data
    - $\eta_k(\boldsymbol{x}) := P(Y = k \mid \boldsymbol{X} = \boldsymbol{x})$, $\boldsymbol{\eta}(\boldsymbol{x}) = (\eta_1(\boldsymbol{x}), \ldots, \eta_k(\boldsymbol{x}))^T$
    - If $\boldsymbol{X}$ given, Y follows the following multinomial distribution.
$$Y | \boldsymbol{X} = \boldsymbol{x} \sim \mathrm{Multi}(\boldsymbol{\eta}(\boldsymbol{x}))$$
  - **The objective of the classification problem:**
    Finding a function $f : \mathcal{X} \to \{1, \ldots, K\}$ (decision function) that predicts Y when given new data $\boldsymbol{X}$.

Classification and Estimation of Conditional Probability

- **Learning Method**
  - "Learning" = obtaining decision function f from observed data $\mathcal{D}_n$.
  - Empirical Risk Minimization (ERM) is commonly used.
  - ERM = Minimizing the loss function for observed data.
  - Let l be a loss function and $\mathcal{F}_n$ be a hypothesis space (candidates of the estimator). Then, ERM $\hat{f}_n$ is

$$\hat{f}_n \in \operatorname*{argmin}_{f \in \mathcal{F}_n} \frac{1}{n} \sum_{i=1}^{n} l(Y_i, f(\boldsymbol{X}_i))$$

  - Predicting Y when $\boldsymbol{X}$ is given based on the estimated decision function.
  - Various loss functions are employed.
    - Hinge loss: $l(y, f(x)) := 0 \vee (1 - yf(x))$
    - Logistic loss: $l(y, f(x)) := \log(1 + e^{-yf(x)})$

Classification and Estimation of Conditional Probability

- **Estimation of Conditional Probability**
  - By using the hinge loss, we can only estimate the label.
  - Conditional probabilities are required in many applications.
    - e.g., AI-based disease detection
  - Conditional probabilities are helpful for human decision-making!
  - Estimation of conditional probabilities ⇒ **Logistic regression!**
  - **Nonparametric Maximum Likelihood Estimator (NPMLE)**

    $\hat{\boldsymbol{p}}_n(\boldsymbol{x}) = (\hat{p}_1(\boldsymbol{x}), \ldots, \hat{p}_K(\boldsymbol{x}))$ is defined as

$$\hat{\boldsymbol{p}}_n \in \operatorname*{arg\,min}_{\boldsymbol{p} \in \mathcal{F}_n} l(\boldsymbol{p}), \quad l(\boldsymbol{p}) := -\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{Y}_i^\top \log \boldsymbol{p}(\boldsymbol{X}_i).$$

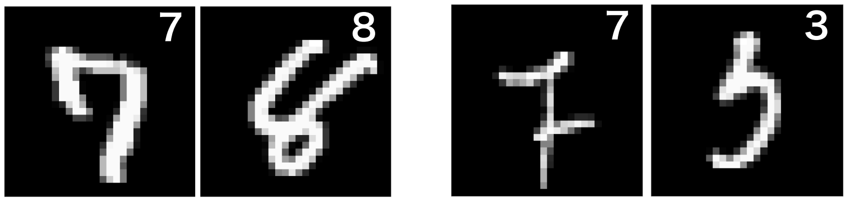
    - $\boldsymbol{Y}_i$ is a one-hot representation of Y_i

      i.e., $\boldsymbol{Y}_i = (0, \ldots, \underset{k}{1}, \ldots, 0)$ if $Y_i = k$

Classification and Estimation of Conditional Probability

- **Advantages of Deep Learning (Deep Neural Networks)**
  - In recent years, it has been shown that, **in the regression problem**, the curse of dimensionality can be mitigated by using DNN models.

    e.g. Shmidt-Hieber (2020), Suzuki (2019)
  - **We can expect this benefit of DNN even for the classification problem!**
  - **Example: MNIST handwriting data (DNN vs kNN)**
    - For images that are difficult to distinguish, DNN provides small conditional probabilities.



DNN	**4: 10%, 7: 90%**	**4: 10%, 8 : 90%**	**7: 60%, 8 : 40%**	**3: 60%, 4: 10%, 5: 30%**
kNN	7: 80%, 9: 20%	4: 30%, 8 : 20%, 9: 50%	1: 100%	3: 40%, 5: 50%, 9: 10%

Semiparametric Estimation and Estimation of Conditional Probability

- In the context of semiparametric estimation, we sometimes need to estimate conditional probabilities.
- In double machine learning (DML; Chernozhukov et al., 2018), we need to show **the specific convergence rate** ($o_p(n^{-1/4})$) **for the nonparametric estimator of the nuisance parameters.**
- When constructing plug-in estimator using an estimator of conditional probability, it is also important that recover the theoretical property of the estimator of conditional probability.

Related Works

Theoretical Study of Classification Problem with Deep Learning

- **Results about the convergence of misclassification rate**
  - misclassification rate = the percentage of times a classifier is incorrect
  $$\mathbb{P}\left(Y \neq \hat{f}_n(\boldsymbol{X})\right)$$
  - misclassification rate is the expectation of 0-1 loss
    - 0-1 loss : $l(y, f(\boldsymbol{x})) = \mathbb{1}\left(y \neq f(\boldsymbol{x})\right)$

    ⇒ we want to minimize 0-1 loss.
  - 0-1 loss is discontinuous, making optimization challenging.

    ⇒ convex surrogate losses are used instead
    - e.g., hinge loss, logistic loss
  - Research is being conducted on the convergence of misclassification rate when using deep learning models with various surrogate loss functions. (Hu et al., 2020; Kim et al., 2021)

Theoretical Study of Classification Problem with Deep Learning

- **Results regarding the convergence of misclassification rate**
  - Hu et al. (2020, arXive:2001.06892)
    - **evaluation metric :** misclassification rate
    - **loss function :** 0-1 loss, hinge loss
    - **assumptions about population distribution :**
      1. The conditional density function of $\boldsymbol{X}$ given Y can be represented by DNN
      2. The probability of data concentrating near the decision boundary is sufficiently low.
  - Kim et al. (2021, Neural Networks)
    - **evaluation metrics :** misclassification rate
    - **loss function :** hinge loss, logistic loss
    - **assumptions about population distribution :**
      1. smooth decision boundary
      2. smooth true conditional probability
      3. margin condition: The density of input $\boldsymbol{X}$ is low near the decision boundary.

Theoretical Study of Classification Problem with Deep Learning

- **Results about the convergence of conditional probability**
  - In classification problems using DNNs, you can output the conditional probabilities for each class by using the softmax function in the final layer.
  - Natural evaluation metric : The difference between the expectation of negative log-likelihood of the estimator and the true conditional probability
  $$\mathbb{E}_{\boldsymbol{X}}\left[\mathrm{KL}\left(\boldsymbol{\eta}(\boldsymbol{X}) \,\|\, \hat{\boldsymbol{p}}(\boldsymbol{X})\right)\right] \quad (*)$$
    - KL : Kullback-Leibler divergence
  - The KL divergence can easily diverge, making it difficult to evaluate (*).
    - If $\mathcal{F}_n$ contains all piecewise constant conditional class probabilities with at most two pieces or all piecewise linear conditional class probabilities with at most three pieces.

      ⇒ (*) diverge (Lemma 2.1 in Bos & Schmidt-Hieber, 2022)
  - **Various approaches have been proposed to address this issue. (Ohn & Kim, 2022; Bos & Schmidt-Hieber, 2022)**

Theoretical Study of Classification Problem with Deep Learning

- **Results about the convergence of conditional probability**
  - Ohn and Kim (2022, Neural computation)
    - **evaluation metric :** KL divergence
    - **loss function :** logistic loss
    - **assumptions about population distribution :** The true conditional probability is bounded away from 0 and 1.
  - Bos & Schmidt-Hieber (2022, Electronic Journal of Statistics)
    - **evaluation metric :** truncated KL divergence
    - **loss function :** negative log-likelihood
    - **assumptions about population distribution :** The true conditional probability belongs to the class $\mathcal{H}$, which is α-SVB.

$$\exists C > 0,\ \forall \boldsymbol{p} \in \mathcal{H};\ \mathbb{P}_{\boldsymbol{X}}\left(p_k(\boldsymbol{X}) \geq t\right) \leq Ct^{\alpha}$$
$$\text{for all } t \in (0,1] \text{ and all } k \in \{1, \ldots, K\}$$

Remained Problems

- There is limited discussion on the theoretical properties of non-parametric estimation of conditional probabilities when the estimator lacks an explicit representation.
- Ohn & Kim (2022) : Results in limited situations.
- Bos & Schmidt-Hieber (2022) : By evaluating the truncated KL divergence instead of KL divergence, assumptions were removed to demonstrate a slightly weaker form of convergence.
  - There has been no discussion on the optimality of the derived rates.
- **This work**
  - **Theoretical investigation of the properties of the NPMLE in classification problems.**
  - **Deriving the rate of convergence of the NPMLE in deep learning.**
  - **Prooving minimax optimality.**

Summary of the work

- **Evaluating KL divergence can be challenging in NPMLE.**
  e.g., in Ohn & Kim (2022), strong assumptions are required.
  ⇒ Resolved the issue by employing the approach of van de Geer (2000) to evaluate the Hellinger distance directly. (Theorem 1, 2)
  - **No strong assumptions are required.**
  - **There is a good insight into the proof.**
- **Application of the theoretical analysis of logistic regression with deep learning. (Theorem 3)**
  - **Avoiding the curse of dimensionality.**
- **Proved that the derived convergence rate is minimax optimal. (Theorem 4)**
  - **It is understood that classification problems are more challenging than regression problems.**

Main Results

Notations and Assumptions

- **Notations**
  - $\bar{\mathcal{F}}_n^{1/2}(\tilde{\boldsymbol{p}},\delta) = \left\{\frac{\boldsymbol{p}+\tilde{\boldsymbol{p}}}{2} : R\left(\frac{\boldsymbol{p}+\tilde{\boldsymbol{p}}}{2},\tilde{\boldsymbol{p}}\right) \le \delta^2,\ \boldsymbol{p}\in\mathcal{F}_n\right\}$
  - $N_{p,B}(\delta,\mathcal{F},Q)$:

    δ-bracketing number of function class $\mathcal{F}$ for $L^p(Q)$ metric
    - Representing a certain "size" or "complexity" of a function class $\mathcal{F}$.
  - $J_B(\delta,\bar{\mathcal{F}}_n^{1/2}(\tilde{\boldsymbol{p}}_n,\delta),\mu) = \int_{\delta^2/(2^{13}c_0)}^{\delta}\sqrt{\log N_{2,B}\left(u,\bar{\mathcal{F}}_n^{1/2}(\tilde{\boldsymbol{p}}_n,\delta),\mu\right)}\,du \vee \delta$
    - μ is the product measure of a counting measure on $\{1,\ldots,K\}$ and the P_X.
    - $J_B(\delta,\bar{\mathcal{F}}_n^{1/2}(\tilde{\mathbf{p}}_n,\delta),\mu)$ represents local complexity around $\tilde{\boldsymbol{p}}_n$.

    ⇒ Using local complexity allows for deriving tight inequalities.

    (Bartlett, 2005)

Notations and Assumptions

- **Notations**
  - $H(P,Q)^2 := \frac{1}{2}\int\left(\sqrt{dP}-\sqrt{dQ}\right)^2$: squared Hellinger distance
  - $R(\boldsymbol{\eta}(\boldsymbol{X}),\boldsymbol{p}(\boldsymbol{X})) := \mathbb{E}_{\boldsymbol{X}}\left[H^2(\boldsymbol{\eta}(\boldsymbol{X}),\boldsymbol{p}(\boldsymbol{X}))\right]$
    - In this study, we discuss the convergence of $R(\boldsymbol{\eta}(\boldsymbol{X}),\hat{\boldsymbol{p}}(\boldsymbol{X}))$ instead of $\mathbb{E}_{\boldsymbol{X}}[\mathrm{KL}(\boldsymbol{\eta}(\boldsymbol{X})\parallel\hat{\boldsymbol{p}}(\boldsymbol{X}))]$.
    - $R(\boldsymbol{\eta}(\boldsymbol{X}),\hat{\boldsymbol{p}}_n(\boldsymbol{X})) \le \mathbb{E}_{\boldsymbol{X}}[\mathrm{KL}(\boldsymbol{\eta}(\boldsymbol{X})\parallel\hat{\boldsymbol{p}}_n(\boldsymbol{X}))]$
- **Assumptions**
  - $\exists\tilde{\boldsymbol{p}}_n\in\mathcal{F}_n,\ \exists c_0,\ \forall\boldsymbol{x}\in\mathcal{X};\ \frac{\eta_k(\boldsymbol{x})}{\tilde{p}_k(\boldsymbol{x})}\le c_0^2,\ k=1,\ldots,K$
    - The only assumption imposed in this study.
    - If $\exists c>0,\ \forall\boldsymbol{p}\in\mathcal{F}_n;\ p_k\ge c,\ k=1,\ldots,K$ then the assumption is satisfied.
    - Automatically satisfied with appropriately chosen network parameters in deep learning.

Oracle Inequality

☆**Oracle Inequality**

Theorem 1 (Evaluation of Variance)

Consider the K class classification problem. Let $\hat{\boldsymbol{p}}_n$ be a NPMLE. Take $\Psi(\delta)\ge J_B(\delta,\bar{\mathcal{F}}_n^{1/2}(\tilde{\boldsymbol{p}}_n,\delta),\mu)$ in such a way $\Psi(\delta)/\delta^2$ is a non-increasing function of δ. Then for universal constant c and for

$$\sqrt{n}\delta_n^2 \ge c\Psi(\delta_n), \qquad (1)$$

we have for all $\delta\ge\delta_n$,

$$\mathbb{P}\left(R(\hat{\boldsymbol{p}}_n,\boldsymbol{\eta}) > 514(1+c_0^2)(\delta^2+R(\tilde{\boldsymbol{p}}_n,\boldsymbol{\eta}))\right)\le c\exp\left(-\frac{n\delta^2}{c^2}\right)$$

- According to Theorem 1, if the hypothesis space $\mathcal{F}_n$ is defined, and its local complexity J_B can be computed, then by solving the inequality (1) (the critical inequality), we can find δ_n.
- Assuming a function class to which $\boldsymbol{\eta}$ belongs, and evaluating the approximation error $R(\tilde{\boldsymbol{p}}_n,\boldsymbol{\eta})$, we can determine rate of convergence.

Oracle Inequality

☆**Oracle Inequality**

Theorem 2 (oracle inequality)

Under the conditions of Theorem 1, we have for some universal constant c,

$$\mathbb{E}\left(R(\hat{\boldsymbol{p}}_n,\boldsymbol{\eta})\right)\le 514(1+c_0^2)\left(\delta_n^2+R(\tilde{\boldsymbol{p}}_n,\boldsymbol{\eta})\right)+\frac{c}{n}$$

- In Theorem 2, δ_n represents the variance of the estimator, while $R(\tilde{\boldsymbol{p}}_n,\boldsymbol{\eta})$ represents the bias of the estimator.
- This inequality is the expected value version of Theorem 1's inequality.

Rate of Convergence in Deep Learning

- **Settings of hypothesis space**
  - $\sigma(x) := \max(0, x)$: activation function, $\mathbf{v} = (v_1, \ldots, v_r) \in \mathbb{R}$: bias
  - $\sigma_{\mathbf{v}}(y_1, \ldots, y_r) = (\sigma(y_1 - v_1), \ldots, \sigma(y_r - v_r))$: activation with bias
  - L : depth of network, $\boldsymbol{m} = (m_0, \ldots, m_{L+1})$: width of network
  - $W_j \in \mathbb{R}^{m_{j+1} \times m_j}$: weight matrix, $\boldsymbol{v}_i \in \mathbb{R}^{m_i}$: bias vector
  - $\boldsymbol{\Phi}$: softmax function
  - Feedforward neural network (FFNN) :

$$f : \mathbb{R}^{m_0} \to \mathbb{R}^{m_{L+1}}, \ \boldsymbol{x} \mapsto f(\boldsymbol{x}) = \boldsymbol{\Phi} W_L \sigma_{\boldsymbol{v}_L} W_{L-1} \sigma_{\boldsymbol{v}_{L-1}} \cdots W_1 \sigma_{\mathbf{v}_1} W_0 \boldsymbol{x} \quad (\star)$$

  - $\mathcal{F}(L, \boldsymbol{m}) := \left\{ \text{f of the form } (\star) : \max_{j=0,\ldots,L} \|W_j\|_\infty \vee |\boldsymbol{v}_j|_\infty \leq 1 \right\}$
  - $\|W_j\|_0$: the number of non-zero entries of W_j
  - $|\boldsymbol{v}_j|_0$: the number of non-zero entries of $\boldsymbol{v}_j$
  - $\mathcal{F}(L, \boldsymbol{m}, s) := \left\{ f \in \mathcal{F}(L, \boldsymbol{m}) : \sum_{j=0}^{L} (\|W_j\|_0 + |\boldsymbol{v}_j|_0 \leq s) \right\}$

Rate of Convergence in Deep Learning

- **Settings of underlying true conditional probability**
  - Hölder space with smoothness β :

$$\mathcal{C}^\beta(D, Q) = \left\{ f : D \subset \mathbb{R}^m \to \mathbb{R} : \sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|<\beta} \|\partial^{\boldsymbol{\alpha}} f\|_\infty + \sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|=\lfloor\beta\rfloor} \sup_{\boldsymbol{x},\boldsymbol{y}\in D,\ \boldsymbol{x}\neq\boldsymbol{y}} \frac{|\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x}) - \partial^{\boldsymbol{\alpha}} f(\boldsymbol{y})|}{|\boldsymbol{x} - \mathbf{y}|_\infty^{\beta - \lfloor\beta\rfloor}} \leq Q \right\}$$

  - Composition Structured Functions :

$$\mathcal{G}_{\text{comp}}(r, \mathbf{d}, \mathbf{t}, \boldsymbol{\beta}, Q) = \{ f = \boldsymbol{g}_r \circ \cdots \circ \boldsymbol{g}_0 : \boldsymbol{g}_i = (g_{ij})_j : [a_i, b_i]^{d_i} \to [a_{i+1}, b_{i+1}]^{d_{i+1}}, \ g_{ij} \in \mathcal{C}^{\beta_i}([a_i, b_i]^{t_i}, Q), \text{for some} |a_i|, |b_i| \leq Q \}$$

    - $f(x_1, x_2, x_3) = g_1 \circ \boldsymbol{g}_2(x_1, x_2, x_3) = g_{11}(g_{01}(x_1, x_3), g_{02}(x_1, x_2))$

      is an example when $d_0 = 3, t_0 = 2, d_1 = t_1 = 2, d_2 = 1$.

  - $\eta_k \in \mathcal{G}_{\text{comp}}(r, \mathbf{d}, \mathbf{t}, \boldsymbol{\beta}, Q), \quad k = 1, \ldots, K$
    - In cases where the dimensions become significantly smaller in each compositional step ($t_i \leq d_i$), it is sufficient to consider t_i dimensions.

      ⇒ avoid the curse of dimensionality.

Rate of Convergence in Deep Learning

☆Rate of Convergence in Deep Learning

- $\mathcal{F}_n = \mathcal{F}(L, \boldsymbol{m}, s)$

Theorem 3 (Rate of Convergence in Deep Learning)

Coonsider the K class classification problem. Let $\hat{\boldsymbol{p}}_n$ be a NPMLE taking values in the network class $\mathcal{F}(L, \boldsymbol{m}, s)$ satisfying

$$\text{(i)} n\phi_n \lesssim \min_{i=1,\ldots,L} m_i, \ \text{(ii)} s \asymp n\phi_n \log(n), \ \text{(iii)} L \asymp \log(n).$$

Then, there exists constant C only depending on $r, d, \boldsymbol{t}, \boldsymbol{\beta}$ such that

$$\mathbb{E}\left[R\left(\hat{\boldsymbol{p}}_n, \boldsymbol{\eta}\right)\right] \leq C\phi_n L \log^2(n),$$

where $\beta_i^* = \beta_i \prod_{l=i+1}^{r} (\beta_l \wedge 1), \ \phi_n = \max_{i=0,\ldots,r} K^{\frac{\beta_i^* + 3t_i}{\beta_i^* t_i}} n^{-\frac{\beta_i^*}{\beta_i^* + t_i}}$

- The rate of convergence is ϕ_n up to log-factor.
- **The rate of convergence is not dependent on dimension d.**

  (avoiding the curse of dimensionality)

Rate of Convergence in Deep Learning

☆Minimax Optimality

Theorem 4 (minimax optimality)

Consider the K class classification problem with X_j drawn from a distribution with Lebesgue density on $[0,1]^d$ which is lower and upper bounded by positive constants. For any nonnegative integer r, any dimension vectors $\boldsymbol{d}$ and $\boldsymbol{t}$ satisfying $t_i \leq \min(d_0, \ldots, d_{i-1})$ for all i, any smoothness vector $\boldsymbol{\beta} \in \mathbb{R}^r$ and all sufficiently large constants $Q > 0$, there exists a positive constant c such that

$$\inf_{\hat{\boldsymbol{p}}_n} \sup_{\eta_k \in \mathcal{G}_{\text{comp}}(r, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{\beta}, Q),\ k=1,\ldots,K} \mathbb{E}\left[R\left(\hat{\boldsymbol{p}}_n, \boldsymbol{\eta}\right)\right] \geq c\phi_n.$$

- According to Theorem 4, the rate of convergence derived in Theorem 3 is minimax optimal up to log-factor.

Rate of Convergence in Deep Learning

- **Comparison with regression**
  - Consider regression.
  - The true regression function f satisfies $f \in \mathcal{G}_{\mathrm{comp}}(r, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{\beta}, Q)$
  - The rate of convergence in regression : $\max_{i=1,\ldots,r} n^{-\frac{2\beta_i^*}{(2\beta_i^*+t_i)}}$
    - minimax optimal (Schmidt-Hieber, 2020)
  - The rate of convergence in classification : $\max_{i=1,\ldots,r} n^{-\frac{\beta_i^*}{\beta_i^*+t_i}}$
  - Regression achieves faster rate.

    ⇒Suggesting that estimating conditional probabilities from given covariates and labels is a more challenging problem than estimating regression functions.
  - **Remark**
    - If the true conditional probability is bounded away from 0 and 1, classification archives the same rate as regression.
    - The true function class can be arbitrarily changed and analyzed as long as it allows for bias evaluation, not limited to this example.

Rate of Convergence in Deep Learning

- **Comparison with Related Work**

	evaluation metrics	Loss function	Remarks
Hu et al. (2020)	misclassification rate	0-1 loss hinge loss	assume strong condition **minimax optimal**
Kim et al. (2021)	misclassification rate	hinge loss logistic loss	assume strong condition **minimax optimal**
Ohn & Kim (2022)	**estimating conditional probability** KL divergence	logistic loss	assume strong condition with regularization
Bos & Schmidt-Hieber (2022)	**estimating conditional probability** truncated KL divergence	negative log-likelihood	general condition
This work	**estimating conditional probability** **Hellinger distance**	**negative log-likelihood**	**general condition** **minimax optimal**

Reference

Sara A Geer, Sara van de Geer, and D Williams. Empirical Processes in M-estimation, volume 6. Cambridge university press, 2000.

Evarist Giné and Richard Nickl. Mathematical Foundations of Infinite-Dimensional Statistical Models, volume 40. Cambridge University Press, 2016.

Yongdai Kim, Ilsang Ohn, and Dongha Kim. Fast convergence rates of deep neural networks for classifi-cation. Neural Networks, 138:179–197, 2021.

Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with ReLU activation function. The Annals of Statistics, 48(4):1875 – 1897, 2020.

Taiji Suzuki. Adaptivity of deep ReLU network for learning in Besov and mixed smooth Besov spaces: optimal rate and curse of dimensionality. In International Conference on Learning Representations, 2019.

Martin J Wainwright. High-dimensional statistics: A non-asymptotic viewpoint, volume 48. Cambridge University Press, 2019.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, James Robins. Double/debiased machine learning for treatment and structural parameters. The Econometrics Journal, Volume 21, Issue 1, 1 February 2018, Pages C1–C68

Reference

Imaizumi, M., & Fukumizu, K. (2019). Deep neural networks learn non-smooth functions effectively. In Proceedings of the International Conference on Artificial Intelligence and Statistics.

Ilsang Ohn, Yongdai Kim; Nonconvex Sparse Regularization for Deep Neural Networks and Its Optimality. Neural Comput 2022; 34 (2): 476–517.